Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Застосування згорткових нейронних мереж у задачі виявлення об'єктів
(тема)

Виконав:
студент 2 курсу, групи СШІм-18-3
спеціальності 122 – Комп'ютерні науки
(код і повна назва спеціальності)
Системи штучного інтелекту (СШІ)
(повна назва спеціалізації)
Березовський Г. В.
(прізвище, ініціали)

Керівник
проф. каф. ШІ Кулішова Н. Є.
(посада, прізвище, ініціали)

Допускається до захисту

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

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

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

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

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

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Березовському Георгію В'ячеславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Застосування згорткових нейронних мереж у задачі виявлення об'єктів.

затверджена наказом університету від 30 березня 2020 р. № 480Ст

2. Термін подання студентом роботи до екзаменаційної комісії 21 травня 2020 р.

3. Вихідні дані до роботи Зображення реальних сцен, де присутній текст, побудований з різних шрифтів, орієнтований вздовж різних напрямків. Набір даних SynthText обсягом в 800 тисяч зображень, фреймворк розробки глибинних нейронних мереж PyTorch

4. Перелік питань, що потрібно опрацювати в роботі Вступ, 1 Нейронно-мережевий підхід для виявлення тексту, Постановка задачі дослідження, 2 Особливості підходу CRAFT до виявлення текстових регіонів в реальних сценах, 3 Нейронно-мережевий підхід YOLO для виявлення тексту, 4 Комбінований нейронно-мережевий підхід для виявлення тексту, 5 Експериментальне дослідження ефективності комбінованого підходу до виявлення

областей тексту в реальних сценах, Висновки, Перелік посилань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_ Мета, основна задача, обєкт та предмет дослідження; Схематичне зображення мережевої архітектури CRAFT, Схема процедури розділення символів, Схема роботи нейронної мережі YOLO, Архітектура YOLO, Візуалізація тренування мережі з функцією втрат DIoU, Результати роботи CRAFT мережі, Приклад вигнутих текстів в наборі даних TotalText, Розподіл помилок, Криві точності-відкликання на наборі даних Picasso, Порівняння результатів виявлення тексту використанням мереж: а) CRAFT; б) комбінованого підходу.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основний	проф. каф. ШІ Кулішова Н. Є.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	30.03.2020	виконано
2	Аналіз предметної галузі	04.04.2020	виконано
3	Постановка задачі дослідження	09.04.2020	виконано
4	Вивчення літератури, дослідження нейромережевих підходів до виявлення текстів у зображеннях	14.04.2020	виконано
5	Дослідження підходу CRAFT до виявлення тексту	20.04.2020	виконано
6	Дослідження підходу YOLO до виявлення тексту	25.04.2020	виконано
7	Розробка комбінованої архітектури до виявлення тексту у зображеннях	1.05.2020	виконано
8	Аналіз вхідних даних та їх підготовка для моделі	3.05.2020	виконано
9	Експериментальне дослідження розробленої архітектури	10.05.2020	виконано
10	Оформлення пояснювальної записки	12.05.2020	виконано
11	Підготовка графічного матеріалу	15.05.2020	виконано
12	Захист атестаційної роботи	21.05.2020	виконано

Дата видачі завдання 30 березня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. ШІ, к.т.н., доц. Кулішова Н. Є.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 73 с., 20 рисунків, 8 таблиць, 62 джерел.

ВИЯВЛЕННЯ ОБ’ЄКТІВ, КОНВОЛЮЦІЙНІ НЕЙРОННІ МЕРЕЖІ, ВИЯВЛЕННЯ ТЕКСТОВИХ СЦЕН, ГЛИБИННЕ НАВЧАННЯ, CRAFT, YOLO

Метою атестаційної роботи є підвищення точності автоматичного виявлення областей тексту на природних зображеннях.

Об'єктом дослідження є задача виявлення тексту на зображеннях.

Предметом дослідження є нейромережеві підходи та алгоритми в задачі виявлення тексту на зображеннях реальних сцен.

В атестаційній роботі розглядаються основні сучасні глибинні нейронні мережі, які використовуються для виявлення на зображеннях об’єктів взагалі та тексту зокрема. В роботі проводиться порівняння архітектур глибинних мереж, аналіз їх недоліків. Розроблено комбіновану архітектуру для ефективного виявлення тексту у природних зображеннях. Проведено експериментальне дослідження дії комбінованої архітектури.

РЕФЕРАТ

Записка пояснительная: 73 с., 20 рисунков, 8 таблиц, 62 источника.

ОБНАРУЖЕНИЕ ОБЪЕКТОВ, СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, ОБНАРУЖЕНИЕ ТЕКСТОВЫХ СЦЕН, ГЛУБОКОЕ ОБУЧЕНИЕ, CRAFT, YOLO

Целью атестационной работы является увеличение точности автоматического обнаружения областей текста на природных изображениях.

Объектом исследования является задача обнаружения текста на изображениях.

Предметом исследования являются нейросетевые подходы и алгоритмы в задаче обнаружения текста на изображениях реальных сцен.

В атестационной работе рассматриваются основные современные глубокие нейронные сети, которые используются для обнаружения на изображениях объектов в общем и текста в частности. В работе проводится сравнение архитектур глубоких сетей, анализ их недостатков. Разработана комбинированная архитектура для эффективного обнаружения текста в природных изображениях. Проведены экспериментальные исследования действия комбинированной архитектуры.

ABSTRACT

Explanatory note: 73 pages, 20 figures, 8 tables, 62 sources.

OBJECT DETECTION, CONVOLUTIONAL NEURAL NETWORKS, SCENE TEXT DETECTION, DEEP LEARNING, CRAFT, YOLO

The purpose of thesis is to increase the accuracy of automatic scene text detection.

The object of study is the task of scene text detection.

The subject of research is neural network approaches and algorithms in the problem of scene text detection.

This thesis examines the main modern deep neural networks for object detection and scene text detection. This thesis compares architectures of deep neural networks, analyzes their shortcomings. A combined architecture was developed for efficient scene text detection. An experimental study of the combined architecture was conducted.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CRAFT — Character Region Awareness For scene Text detection (обізнаність на рівні символів для виявлення тексту в зображеннях реальних сцен);

YOLO — You Only Look Once (однопрохідна нейронно-мережева архітектура для виявлення об’єктів);

IOU — Intersection Over Union (метрика для оцінки схожості між прямокутними обмежувальними полями);

DIOU — Distance Intersection Over Union (модифікована IOU метрика);

ICDAR — International Conference on Document Analysis and Recognition (Міжнародна конференція з аналізу та розпізнавання документів);

MSRA-TD500 — MSRA Text Detection Database (набір даних MSRA для виявлення тексту);

CTW-1500 — Curve Text in the Wild (набір даних для виявлення кривого тексту);

MSER — Maximally Stable Extremal Regions (алгоритм вилучення ознак з зображень);

SWT — Stationary Wavelet Transform (стаціонарна вейвлет-трансформація);

SSD — Single Shot Detector (однопрохідна нейронно-мережева архітектура для виявлення об’єктів);

R-CNN — Region Convolutional Neural Network (багатопрохідна нейронно-мережева архітектура для виявлення об’єктів);

DMPNet — Deep Matching Prior Network (мережа глибокого узгодження)

FCN — Fully Convolutional Network (нейронно-мережева архітектура, в якій немає повнозв’язних слоїв);

DPM — Deformable Part Model (детектор об’єктів);

RSSD — Rotation-sensitive regression for oriented scene text detection (чутливий до обертання детектор тексту).

ВСТУП

У 21 столітті з розвитком високошвидкісного інтернету, технологій і підходів задля поліпшення людської праці і життєдіяльності завдяки методам штучного інтелекту, а також із глобальним трендом до активного застосування комп'ютерного зору стало дуже актуальним завдання виявлення тексту реальних сцен. Широкий спектр додатків, від розпізнавання номерних знаків до безпілотних автомобілів, говорить про велику затребуваність даної теми.

Виявлення та розпізнавання сценічного тексту може допомогти безпілотному автомобілю розрізняти дорожні покажчики, вивіски на будівлях та інше, що набагато підвищить надійність і ефективність використання технології безпілотного транспорту.

Також це виявиться корисним в рамках доповненої реальності, дозволяючи надавати не тільки візуальне уявлення навколишнього світу, а також його семантичну складову.

Методи виявлення тексту в реальних сценах, засновані на нейронних мережах, з’явилися нещодавно і показали багатообіцяючі результати. Алгоритми, котрі вчать. У цій роботі пропонується підхід до виявлення сценічного тексту для ефективного виявлення області тексту шляхом детального вивчення кожного символу та спорідненості між символами, а також подальшого опрацювання символьної інформації блоком загального детектору об’єктів.

1 НЕЙРОННО-МЕРЕЖЕВИЙ ПІДХІД ДЛЯ ВИЯВЛЕННЯ ТЕКСТУ

1.1 Складнощі у виявленні тексту

Виявлення тексту сцени привертає велику увагу до області комп’ютерного зору через численні програми, такі як миттєвий переклад, пошук зображень, аналіз сцени, геолокація та сліпа навігація. Запропоновані нещодавно детектори тексту сцени на основі глибокого навчання показали перспективні показники [1-9]. Ці методи в основному навчають свої мережі для локалізації обмежувальних вікон рівня слова. Однак вони можуть бути не ефективними у важких випадках, таких як вигнуті, деформовані або надзвичайно довгі тексти, які важко виявити за допомогою одного обмежувального поля. Крім того, обізнаність на рівні символів має багато переваг під час роботи зі складними текстами шляхом зв’язування послідовних символів знизу вгору. На жаль, більшість існуючих наборів текстових даних не надають анотацій на рівні символів, а робота, необхідна для отримання анотацій на рівні символів, занадто дорога.

Для розв'язання цієї проблеми авторами [10] був запропонований новий детектор тексту, який локалізує окремі області символів та пов'язує виявлені символи з текстовим інстансом. Підхід, іменований CRAFT (Character Region Awareness For Text) — інформованість регіонів символів для виявлення тексту, розроблений за допомогою звивистої нейронної мережі, яка створює оцінку факту, що регіон є символом, та оцінку достовірності цього факту. Оцінка регіону використовується для локалізації окремих символів на зображенні, а оцінка достовірності використовується для групування кожного символу в одне слово. Щоб компенсувати відсутність анотацій на рівні символів, автори запропонували слабко контрольовану систему навчання, яка оцінює значення достовірності на рівні символів у існуючих реальних наборах даних на рівні слів.

На рис. 1.1 представлено візуалізацію результатів роботи CRAFT на різних текстових формах. Використовуючи інформованість регіону на рівні символів, легко представляються тексти різних форм. Автори демонструють екстенсивні експерименти над наборами даних ICDAR [11] для підтвердження їх методу, а експерименти показують, що запропонований метод перевершує стан сучасних детекторів тексту. Крім того, експерименти з наборами даних MSRA-TD500 [12], CTW-1500 [13] та TotalText [14] показують високу ступінь можливості використання запропонованого способу на складних випадках, таких як довгі, зігнуті та / або тексти довільної форми.

Рисунок 1.1 — Демонстрація ефективності підходу CRAFT

1.2. Сучасні детектори текстів у зображеннях реальних сцен

Основною тенденцією у виявленні тексту сцени до появи глибокого навчання було використання рукописних алгоритмів, які в своїй більшості не підлягають навчанню. В основному використовувались штучні ознаки — наприклад, максимально стійкі екстремальні області (Maximally Stable Extremal Regions - MSER [15]) або трансформація ширини обведення (Stroke Width Transform - SWT [16]) — як основний компонент. Останнім часом були запропоновані глибокі детектори тексту на основі навчання, використовуючи популярні методи виявлення/сегментації об'єктів, такі як однопрохідний детектор орієнтованого тексту (Single-Shot Oriented Scene Text Detector - SSD [17]), швидша згорткова нейронна мережа з пропозиціями регіонів (Faster Region Convolutional Neural Network - Faster R-CNN [18]) та повністю згорткова нейронна мережа (fully convolutional network - FCN [19]).

Ще один підхід реалізують детектори тексту на основі регресії. Запропоновано різні текстові детектори, що використовують регресію обмежувальної коробки, адаптовану з популярних детекторів об'єктів. На відміну від об'єктів взагалі, тексти часто представлені регіоном неправильної форми з різними співвідношеннями сторін. Щоб вирішити цю проблему, TextBoxes [20] модифікував згорткові ядра та anchor boxes, щоб ефективно фіксувати різні форми тексту. Автори мережі глибокого узгодження (Deep Matching Prior Network - DMPNet [21]) намагалися ще більше зменшити проблему, включивши чотирикутні розсувні вікна. Нещодавно було запропоновано детектор регресії, чутливий до обертання (Rotation-sensitive regression for oriented scene text detection - RSDD) [22], який повною мірою використовує ознаки, інваріантні до обертання шляхом активного обертання згорткових фільтрів. Однак, існує структурне обмеження на захоплення всіх можливих фігур, які існують у природі при використанні цього підходу.

Серед іншого, в останні часи інтенсивно розвивається архітектурний підхід до виявлення та аналізу об’єктів на основі нейро-фаззі нейрону [5, 23, 24, 25]. В перспективі цей підхід можливо застосувати й безпосередньо до задачі виявлення тексту.

Інший поширений підхід базується на роботах, що стосуються сегментації, яка має на меті шукати текстові регіони на рівні пікселів. Ці підходи, що виявляють тексти шляхом оцінки областей, що обмежують слова, реалізовано в таких архітектурах, як багатомасштабна повністю згорткова нейронна мережа (Multi-scale Fully Convolutional Neural Network - Multi-scale FCN [26]), цілісне прогнозування (Holistic-prediction [27]) та PixelLink [1], також були запропоновані, використовуючи сегментацію як основу підходу. Автори однопрохідного детектора тексту з регіональною увагою (Single Shot Text Detector with Regional Attention - SSTD [2]) намагалися отримати користь як від регресії, так і від сегментації, використовуючи механізм уваги для розширення області тексту, зменшуючи фонові перешкоди на рівні ознаків. Нещодавно архітектуру TextSnake [7] було запропоновано для виявлення текстових екземплярів, завдяки передбаченню області тексту та центральної лінії разом з атрибутами геометрії.

Підхід «від початку до кінця» одночасно тренує модулі виявлення та розпізнавання, щоб підвищити точність виявлення, використовуючи результат розпізнавання. Автори підходів швидко орієнтованого розміщення тексту за допомогою єдиної мережі (Fast Oriented Text Spotting with a Unified Network - FOTS [6]) та екзистенціальної тривоги знищення (Existential Annihilation Anxieties - EAA [3]) об'єднують популярні методи виявлення та розпізнавання та навчають їх поступово. Автори Mask TextSpotter [8] скористалися їх уніфікованою моделлю, щоб розглянути розпізнавальну задачу як проблему семантичної сегментації. Очевидно, що навчання за допомогою модуля розпізнавання допомагає детектору тексту бути більш стійким до тексту, який схожий на текстові фони.

Більшість методів виявляє текст зі словом як суцільну одиницю, але визначення відрізків слова для виявлення є нетривіальною задачею, оскільки слова можна розділити за різними критеріями, такими як зміст, пробіли чи колір. Крім того, межа сегментації слова не може бути чітко визначена, тому сам сегмент слова не має чіткого змістовного значення. Ця неоднозначність в анотації слів змінює значення основної істини як для підходу до регресії, так і для сегментації.

Відомі також детектори тексту на рівні символів. Zhang та ін. [28] запропонував детектор рівня символів, використовуючи кандидатури текстового блоку, перегнані через модуль максимально стійкого екстремального регіону (Maximally Stable Extremal Region - MSER [15]). Той факт, що він використовує MSER для ідентифікації окремих символів, обмежує стійкість його виявлення в певних ситуаціях, таких як сцени з низькою контрастністю, кривизною та віддзеркаленням світла. Yao та ін. [27] використовують карту передбачення символів разом із картою областей текстових слів та сполученням орієнтацій, які потребують анотацій на рівні символів. Замість явного передбачення рівня символів, Seglink [9] шукає текстові сітки (часткові текстові сегменти) та пов'язує ці сегменти з додатковим передбаченням посилань. Незважаючи на це, Mask TextSpotter [8] передбачає карту ймовірностей рівня символів, вона використовувалася для розпізнавання тексту в цілому замість того, щоб виділяти окремі символи.

Автори підходу CRAFT в своїй роботі натхнені ідеєю WordSup [4], яка використовує слабко контрольовану структуру для тренування детектора рівня символів. Однак недоліком Wordsup є те, що представлення символів формується у прямокутних обмежувальних полях, що робить підхід вразливим до перспективної деформації символів, викликаної різними точками зору камери. Більше того, він пов'язаний з роботою структури кореневої нейронної мережі (тобто, використовуючи SSD), і обмежується кількістю анкерних коробок та їх розмірами).

YOLO — це швидкий, точний детектор об'єктів, що робить його ідеальним для програм комп’ютерного зору. Автори показали, що підключаючи YOLO до веб-камери можна переконатись, що нейронна мережа підтримує продуктивність у режимі реального часу, включаючи час для отримання зображень із камери та відображення виявлених даних.

Отримана система є інтерактивною та привабливою. Хоча YOLO обробляє зображення окремо, при приєднанні до веб-камери він функціонує як система стеження, виявляючи об'єкти, коли вони рухаються навколо та змінюють зовнішній вигляд.

Виявлення об'єктів є основною проблемою в комп'ютерному зорі. Алгоритми виявлення зазвичай починаються з вилучення набору надійних ознак із вхідних зображень (Haar [29], SIFT [30], HOG [31], згорткові особливості). Потім класифікатори [32, 33] або локалізатор [34] використовуються для ідентифікації об'єктів у просторі ознак. Ці класифікатори або локалізатори виконуються або в режимі розсувного вікна над усім зображенням, або над деякою підмножиною регіонів зображення [35]. Далі буде порівняна система виявлення YOLO з кількома основними підходами виявлення, виділяючи ключові подібності та відмінності.

1. Deformable parts models. Моделі деформованих деталей (DPM) використовують підхід розсувного вікна для виявлення об'єктів [32]. DPM використовує роз'єднаний конвеєр для вилучення статичних ознак, класифікації регіонів, прогнозування обмежувальних коробок для регіонів з високим балом тощо. Система YOLO замінює всі ці розрізнені частини однією згорнутою нейронною мережею. Мережа виконує видобуток ознак, прогнозування обмежувальної коробки, non-maximal suppression та контекстуальне міркування. Замість статичних ознак мережа здійснює підготовку ознак в режимі реального часу та оптимізує їх для завдання виявлення. Об'єднана архітектура YOLO призводить до більш швидкої та точної моделі, ніж DPM.

2. R-CNN. R-CNN та його варіанти використовують пропозиції регіонів замість розсувних вікон для пошуку об’єктів у зображеннях. Вибірковий пошук [35] генерує потенційні обмежувальні коробки, згорткова мережа витягує функції, SVM оцінює поля, лінійна модель коригує обмежувальні поля, a non-maximal suppression виключає повторювані виявлення. Кожна стадія цього складного процесу повинна бути точно налаштована незалежно, і отримана система є дуже повільною, займаючи більше 40 секунд на зображення в тестовий час [36].

YOLO поділяє деякі подібності з R-CNN. Кожна клітина сітки пропонує потенційні обмежувальні коробки та проводить їх кількість за допомогою згорткових функцій. Однак авторська система ставить просторові обмеження на пропозиції клітин сітки, що допомагає пом’якшити кілька виявлень одного об'єкта. YOLO також пропонує набагато менше обмежувальних полів, лише 98 на зображення, порівняно з 2000 із селективного пошуку. Нарешті, розглядаєма система поєднує ці окремі компоненти в єдину спільно оптимізовану модель.

Інші швидкі детектори. Fast та Faster R-CNN зосереджується на прискоренні системи R-CNN шляхом обміну обчисленнями та використання нейронних мереж для пропонування регіонів замість селективного пошуку [36, 37]. Хоча вони пропонують покращення швидкості та точності в порівнянні з R-CNN, вони все ще не відповідають швидкісним характеристикам у режимі реального часу.

Багато дослідницьких робіт зосереджуються на прискоренні підходу DPM [38-40]. Вони прискорюють обчислення HOG, використовують каскади та підштовхують обчислення до GPU. Однак система працює в режимі реального часу лише на 30 FPS [39].

Замість того, щоб намагатися оптимізувати окремі компоненти великого процесу для виявлення, YOLO повністю викидає складний процес і є швидким по архітектурі.

Детектори для одиночних класів, як обличчя чи люди, можуть бути оптимізовані, оскільки їм доводиться мати справу зі значно меншими варіаціями [41]. YOLO — детектор загального призначення, який вчиться одночасно виявляти різноманітні об'єкти.

Deep MultiBox. На відміну від R-CNN, Szegedy та ін. навчати конволюційну нейронну мережу для прогнозування регіонів [42], а не використовувати селективний пошук. MultiBox також може виконувати виявлення одного об'єкта, замінивши прогноз достовірності на передбачення для одного класу. Однак MultiBox не може виконувати загальне виявлення об'єктів і все ще є лише частиною у більшій конвеєрній системі виявлення, що вимагає подальшої класифікації патчів зображення. І YOLO, і MultiBox використовують звивисту мережу для прогнозування обмежувальних коробок у зображенні, але YOLO — це повна система виявлення.

OverFeat. Sermanet та ін. навчають конволюційну нейронну мережу для локалізації та адаптації цього локалізатора для виявлення [43]. OverFeat ефективно виконує розпізнавання ковзаючих вікон, але це все ще неперервна система. OverFeat оптимізується для локалізації, а не для виявлення об’єктів. Як і DPM, локалізатор бачить локальну інформацію лише під час прогнозування. OverFeat не може міркувати про глобальний контекст, і тому потрібна значна післяобробка для створення когерентних виявлень.

MultiGrasp. YOLO схожа за дизайном на роботу з виявлення схоплень Redmon et al. [44]. Авторський підхід до прогнозування обмежувальної коробки базується на системі MultiGrasp для регресії до схоплення. Однак виявлення схоплення є набагато простішим завданням, ніж виявлення об'єктів. MultiGrasp потрібно лише передбачити окрему область, що можна зрозуміти для зображення, що воно містить один об'єкт. Не потрібно оцінювати розмір, місцеположення чи межі об’єкта чи передбачати його клас, лише знайти регіон, придатний для розуміння. YOLO прогнозує як обмежувальні поля, так і ймовірності класів для декількох об'єктів декількох класів у зображенні.

YOLO — це універсальна модель для виявлення об'єктів. Вона проста в конструкції, і її можна навчити безпосередньо на повних зображеннях. На відміну від класифікованих підходів, YOLO навчається функції втрат, яка безпосередньо відповідає ефективності виявлення, і вся модель тренується спільно.

Fast YOLO — це найшвидший детектор об'єктів загального призначення в літературі, і YOLO підштовхує розвиток виявлення об'єктів у режимі реального часу. YOLO також добре узагальнює нові домени, що робить його ідеальним для додатків, які покладаються на швидке, надійне виявлення об'єктів.

1.3 Постановка задачі дослідження

Переважно задача виявлення об’єктів на зображенні в цілому, а тексту — зокрема, вирішується двома підходами: однопрохідним та багатопрохідним. В однопрохідному підході нейронна мережа генерує результати виявлення на основі мап ознак, отриманих з основної архітектурної конструкції. Прикладами такого підходу є архітектури YOLO та SSD. Основною перевагою такого підходу є його швидкість та невелика кількість навчальних параметрів та активацій, що робить тренування швидким та стабільним. В багатопрохідному підході на відміну від однопрохідного результати виявлення отримуються з мап ознак, отриманих спеціальним модулем, як то реалізовано в RCNN підходах. Перевагою такого підходу є можливість сегментувати виявлені об’єкти на піксельному рівні.

В задачі виявлення тексту недоліками однопрохідних та багатопрохідних підходів є те, що на рівні архітектури нейронної мережі немає модуля, який був би орієнтований на виявлення безпосередньо символів та зв’язків між ними, що дуже важливо для отримання результатів роботи мережі на рівні координат обмежувальних полей слів. Цю задачу вирішує підхід

CRAFT, але в ньому є необхідність додатково обробляти результати виявлення рукописним алгоритмом, що є слабким місцем цього підходу.

Беручи до уваги сильні та слабкі сторони обох підходів, можна сформулювати постановку задачі дослідження: розробка архітектури глибокої нейронної мережі, яка реалізує комбінований підхід до виявлення тексту в реальних сценах. В рамках комбінованого підходу буде реалізована модель на основі CRAFT та YOLO нейронних мереж.

Для досягнення цієї мети потрібно вирішити декілька окремих задач:

- дослідити архітектуру і особливості роботи мережі CRAFT;
- дослідити архітектуру і принцип дії мережі YOLO;
- розробити основні принципи комбінування глибоких мереж для виявлення текстових регіонів в реальних сценах;
- провести експериментальне дослідження ефективності запропонованого комбінованого підходу.

2 ОСОБЛИВОСТІ ПІДХОДУ CRAFT ДО ВИЯВЛЕННЯ ТЕКСТОВИХ РЕГІОНІВ В РЕАЛЬНИХ СЦЕНАХ

Основна мета підходу CRAFT — точно локалізувати кожний окремий символ в природних образах. З цією метою автори навчають глибоку нейронну мережу для прогнозування регіонів символів та співвідношення між символами. Оскільки немає загальнодоступних наборів даних на рівні загальних символів, модель навчається в слабко контролюємому режимі.

2.1 Архітектура мережі CRAFT

Повністю згорнута мережева архітектура, заснована на VGG-16 [45], з пакетною нормалізацією, прийнята в якості основи. Модель має пропускні з'єднання в частині декодування, що схоже на U-net [46], оскільки вона агрегує ознаки низького рівня. Вихідний висновок має два канали як карти балів: оцінка регіону та оцінка ефективності. Мережева архітектура схематично проілюстрована на рис. 2.1.

2.2 Процес навчання мережі CRAFT

Для кожного навчального зображення автори генерують істинну мітку для виставлення балу регіону та базу достовірності із обмеженими полями рівня знаків. Оцінка регіону представляє ймовірність того, що даний піксель є центром символу, а оцінка достовірності представляє ймовірність центру простору між суміжними символами.

На відміну від бінарної карти сегментації, яка позначає кожен піксель дискретно, автори кодують ймовірність символьного центру за допомогою гаусової теплової карти. Це подання теплової карти було використано і в інших задачах, наприклад, у роботі з оцінки поз [47], через високу гнучкість

теплових карт під час роботи з анотаціями, які жорстко не обмежені. Автори використовують подання теплової карти, щоб визначити показник регіону і показник достовірності.

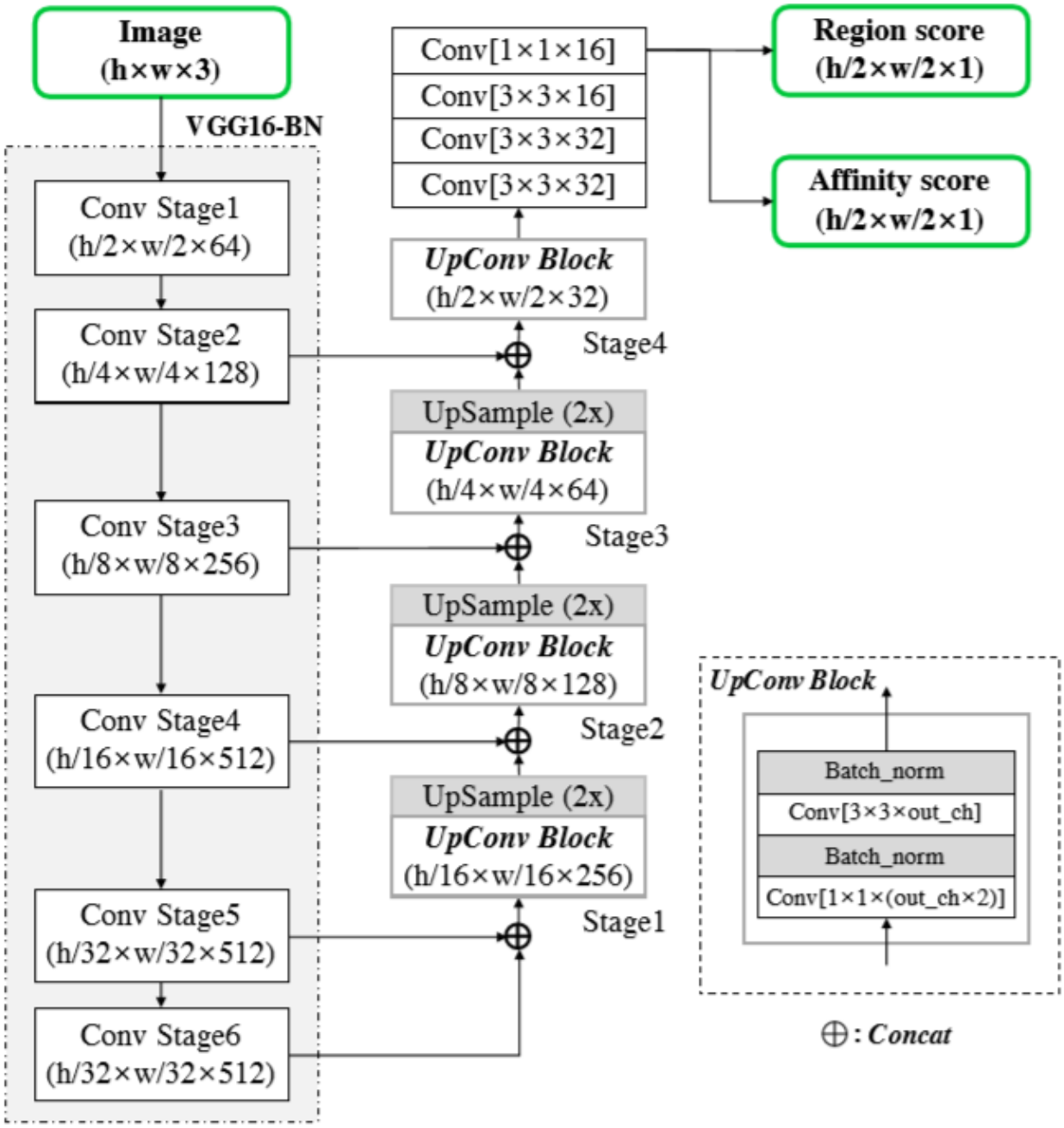


Рисунок 2.1 — Схематичне зображення мережевої архітектури

Рис. 2.2 підсумовує процес генерації міток для синтетичного зображення. Обчислення значення розподілу Гаусса безпосередньо для

кожного пікселя в обмежувальному полі дуже трудомістке. Оскільки вікна, що обмежують символи на зображенні, як правило, спотворюються за допомогою перспективних проекцій, автори використовують наступні кроки, щоб наблизити та генерувати мітку як для балів регіону, так і для оцінки результатів:

1) підготувати двовимірну ізотропну карту Гаусса;

2) обчислити перетворення перспективи між регіоном карти Гаусса та кожним полем символів;

3) викривити карту Гаусса до області коробки.

Для мітки балу достовірності по афінності між полями визначають за допомогою суміжних полів символів, як показано на рис. 2.2. Накресливши діагональні лінії для з'єднання протилежних кутів кожного поля символів, можливо генерувати два трикутники — верхній і нижній трикутники символів. Тоді для кожної суміжної пари символьних коробок формується поле для афінності, встановлюючи центри верхнього та нижнього трикутників як кути поля.

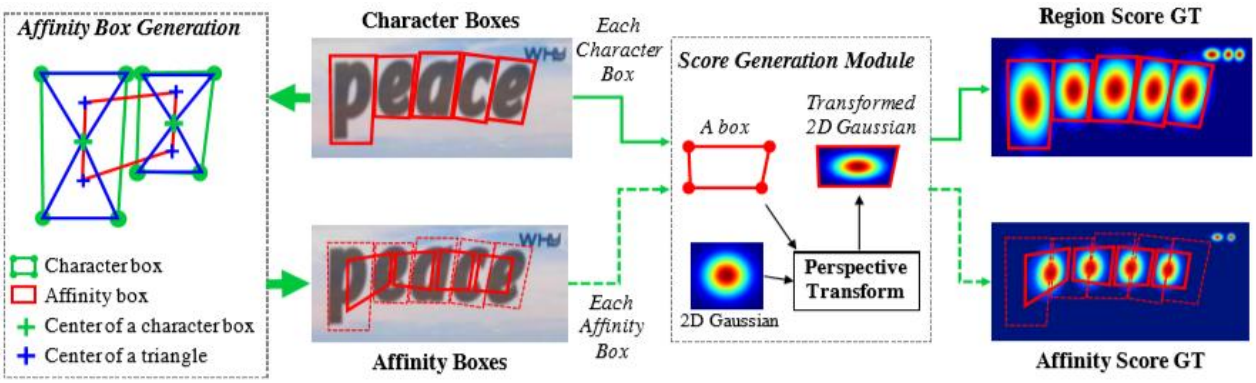


Рисунок 2.2 — Схема генерації істинних міток на рівні символів та їх афінності

Запропоноване визначення істиної мітки дає змогу моделі достатньо виявляти великі або довгі екземпляри тексту, незважаючи на використання

невеликих сприйнятливих полів. З іншого боку, попередні підходи, такі як регресія коробки, потребують значного сприйняття в таких випадках. Авторське виявлення рівня символів дозволяє конволюційним фільтрам зосередитись лише на внутрішньосимвольних та інтер-символьних, а не на всіх текстових екземплярах.

На відміну від синтетичних наборів даних, реальні зображення в наборі даних зазвичай мають анотації на рівні слів. Тут треба генерувати поля символів з анотацій кожного рівня слів під слабким наглядом, як узагальнено на рис. 2.3. Коли надається реальне зображення з анотаціями на рівні слова, вивчена проміжна модель прогнозує бал регіону символів обрізаного слова зображення для створення обмежувальних полів рівня символів. Для того, щоб відобразити надійність прогнозування тимчасової моделі, значення карти зв'язку над кожним полем слів обчислюється пропорційно кількості виявлених символів, поділеному на кількість символів істинної мітки, яке використовується для вагової ваги протягом навчання.

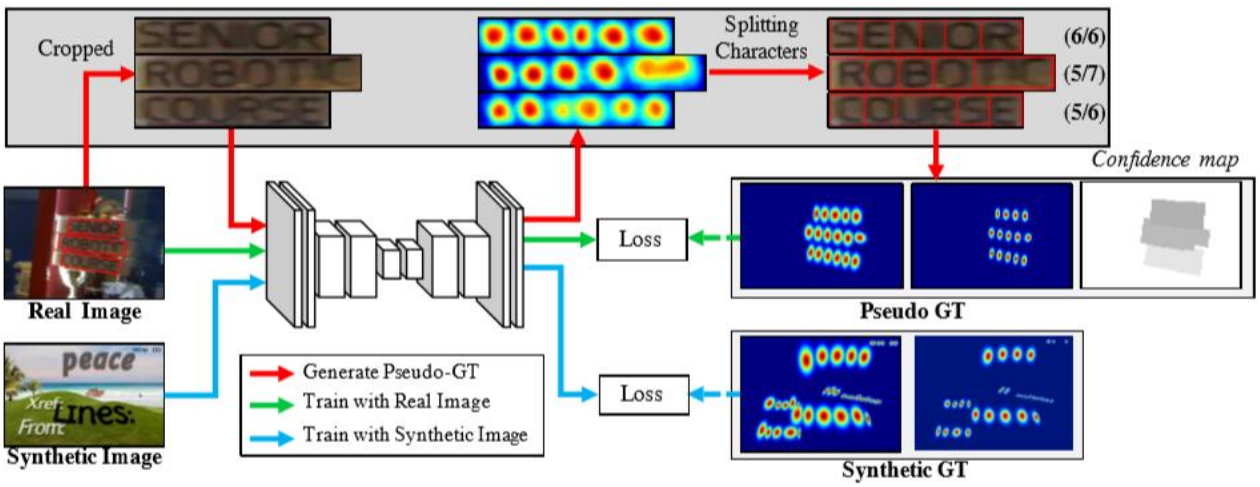


Рисунок 2.3 — Схема тренування CRAFT

На рис. 2.4 показана вся процедура розділення символів. По-перше, зображення на рівні слів обрізаються з вихідного зображення. По-друге,

сучасна модель, підготовлена на той час, прогнозує бал регіону. По-третє, алгоритм Watershed [48] використовується для розділення областей символів, який використовується для виготовлення обмежувальних полів знаків, що охоплюють регіони. Нарешті, координати полів символів перетворюються назад у вихідні координати зображення за допомогою зворотного перетворення на етапі обрізання. Псевдо істинні мітки для оцінки регіону та оцінки достовірності можуть бути згенеровані за допомогою етапів, показаних на рис. 2.2, використовуючи отримані чотирикутні рамки рівня символів.

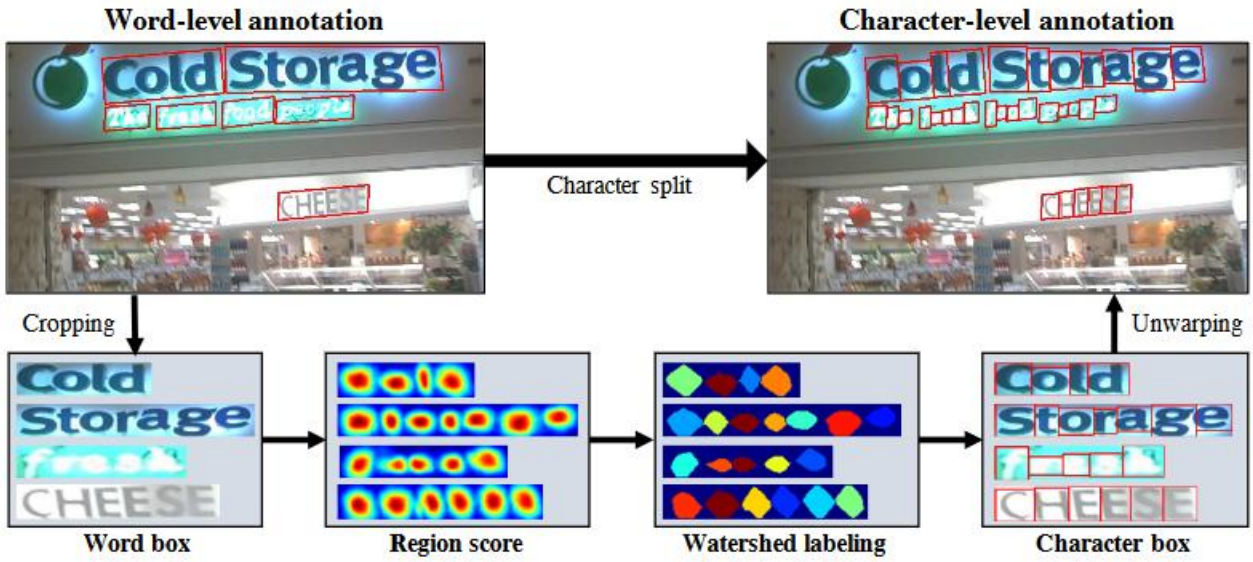


Рисунок 2.4 — Схема процедури розділення символів.

Коли модель тренується за допомогою слабкого нагляду, це змушує тренуватися з неповними псевдо-істинними мітками. Якщо модель підготовлена з неточними балами регіонів, вихід може бути розмитим у регіонах символів. Щоб цього не допустити, приходиться вимірювати якість кожної псевдо-істинох мітки, що генерується моделлю. На щастя, в тексті анотації є дуже сильний сигнал, який є довжиною слова. У більшості наборів даних передбачена транскрипція слів, а довжина слів може бути використана для оцінки впевненості псевдо-істинних міток.

Для анотованого зразка *w* рівня навчальних даних на рівні слова, нехай *R(w)* та *l(w)* — область обмежувального поля та довжина слова вибірки *w* відповідно. За допомогою процесу поділу символів ми можемо отримати оцінені поля обмеження символів та їх відповідну довжину символів *lc(w).* Тоді показник достовірності *s(w)* для вибірки w обчислюється як

$$s_{conf}(w) = \frac{l(w) - \min(l(w), \left| l(w) - l^{C}(w) \right|)}{l(w)}, \tag{2.1}$$

а піксельна карта достовірності для зображення обчислюється як

$$S_c(p) = \begin{cases} s_{conf}(w) & p \in R(w) \\ 1 & otherwise \end{cases} \tag{2.2}$$

де p позначає піксель в області *R(w)*. Функція втрат *L* визначається як

$$L = \sum_{p} S_c(p) \cdot (\left\| S_r(p) - S_r^*(p) \right\|_2^2 + \left\| S_a(p) - S_a^*(p) \right\|_2^2), \tag{2.3}$$

де $S_r^*(p)$ і $S_a^*(p)$ позначають показник області псевдо-істинного балу по регіону так мапу афінності, а $S_r(p)$ і $S_a(p)$ — виявленний мережею бал регіону та мапа афінності. Тренуючись із синтетичними даними, ми можемо отримати справжню основну істину, тому $S_c(p)$ встановлюється рівним 1.

По мірі навчання, модель CRAFT може прогнозувати символи точніше, а результативність $S_{conf}(w)$ також поступово збільшується. На рис. 2.5 показана

карта балів регіону символів під час тренувань. На ранніх етапах навчання бали в регіоні відносно низькі для незнайомого тексту в природних образах. Модель вивчає появу нових текстів, таких як неправильні шрифти, та синтезованих текстів, які мають різний розподіл даних у порівнянні з набором даних SynthText.

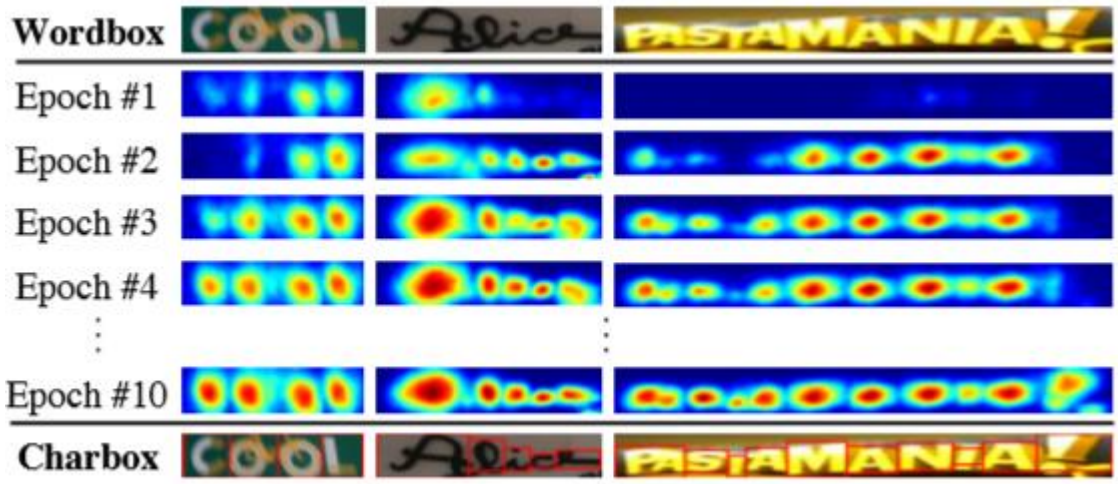


Рисунок 2.5 — Карта балів регіону символів під час тренувань

Якщо показник впевненості $s_{conf}(w)$ нижчий за 0.5, передбачуваними полями обмеження символів слід знехтувати, оскільки вони несуть несприятливі наслідки при навчанні моделі. У цьому випадку припускається, що ширина індивідуального символу є постійною, і обчислюється передбачення рівня символів, просто розділивши область слова *R(w)* на кількість символів *l(w)*. Тоді $s_{conf}(w)$ встановлюється на 0.5, щоб дізнатися небачені досі види текстів.

На етапі тестового часу, остаточний вихід може бути наданий у різних формах, таких як текстові поля або поля символів та інші багатокутники. Для таких наборів даних, як ICDAR, протокол оцінювання є пересічним рівнем перерізування на рівні слова (IoU), тому тут описується, як зробити поле для

обмеження рівня слова QuadBox з передбачуваного S_r та S_a за допомогою простого, але ефективного кроку після обробки.

Післяобробка для фінішних обмежувальних коробок підсумовується наступним чином. По-перше, бінарну карту *M*, що охоплює зображення, ініціалізують на 0. *M(p)* встановлюється на 1, якщо $S_r(p) > \tau_r$ або $S_a(p) > \tau_a$, де τ_r — поріг області, а τ_a — поріг границі. По-друге, проводиться маркування підключеного компонента (Connected Component Labeling - CCL) на *M*. Нарешті, QuadBox отримують шляхом додавання обертового прямокутника з мінімальною площею, що охоплює з'єднані компоненти, що відповідають кожній з міток. Для цього можна застосувати такі функції, як connectedComponents та minAreaRect, що надаються OpenCV.

Слід зауважити, що перевага CRAFT полягає в тому, що він не потребує додаткових методів післяобробки, таких як non-maximal suppression (NMS). Оскільки наявні елементи зображення з областей слів, розділених CCL, обмежувальне поле для слова просто визначається одним прямокутником, що вкладається. З іншого боку, авторський процес зв’язування символів проводиться на рівні пікселів, що відрізняється від інших методів, що базуються на зв'язках [4, 9], чітко покладаючись на пошук відносин між текстовими компонентами.

Крім того, можливо генерувати багатокутник по всій області символів, щоб ефективно боротися з вигнутими текстами. Процедура генерації багатокутника проілюстрована на рис. 2.6. Першим кроком є визначення локальної лінії максимумів символьних областей вздовж напрямку сканування, як показано на зображенні зі стрілками синього кольору. Довжини локальних прямих максимумів однаково задаються як максимальна довжина між ними, щоб запобігти нерівномірності результату кінцевого багатокутника. Лінія, що з'єднує всі центральні точки локальних максимумів, називається центральною лінією, показаною жовтим кольором. Тоді локальні лінії

максимумів повертаються перпендикулярно до центральної лінії, щоб відобразити кут нахилу символів, виражений червоними стрілками. Кінцевими точками локальних максимумів рядків є кандидати на контрольні точки текстового багатокутника. Щоб повністю охопити область тексту, автори переміщують дві зовнішні найбільш нахилені локальні лінії максимуму назовні по локальній центральній лінії максимуму, роблячи кінцеві контрольні точки (зелені точки).

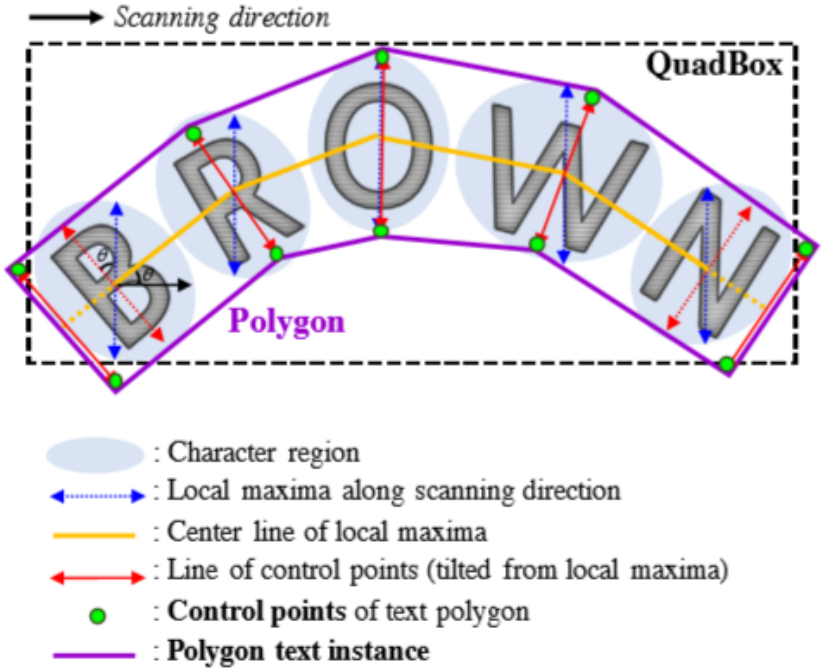


Рисунок 2.6 — Процедура генерації багатокутника

Отже, детектор тексту CRAFT може виявляти окремі символи, навіть коли анотації на рівні символів не надані. Запропонований метод забезпечує оцінку області символів та оцінку достовірності символів, які разом повністю покривають різні текстові фігури знизу вгору. Оскільки реальні набори даних, що надаються з анотаціями на рівні символів, є рідкісними, до архітектури застосовується слабко контрольований метод навчання, який генерує псевдо-ґрунтові анотації з проміжної моделі. CRAFT демонструє найсучасніші

показники точності виявлення на більшості публічних наборів даних та демонструє здатність до узагальнення, показуючи ці показники без налагодження на інших наборах даних. На рис. 2.7 можна побачити результати роботи CRAFT на наборы даних TotalText.

Рисунок 2.7 — Результати роботи CRAFT мережі

3 НЕЙРОННО-МЕРЕЖЕВИЙ ПІДХІД YOLO ДЛЯ ВИЯВЛЕННЯ ТЕКСТУ

Люди дивляться на зображення і миттєво знають, які об’єкти є на зображенні, де вони знаходяться та як вони взаємодіють. Зорова система людини швидка і точна, що дозволяє нам виконувати складні завдання, такі як водіння з мало усвідомленою думкою. Швидкі, точні алгоритми виявлення об'єктів дозволять комп’ютерам керувати автомобілями без спеціалізованих датчиків, дозволять допоміжним пристроям передавати інформацію про місця в реальному часі користувачам людей та розкривати потенціал загальноприйнятих робототехнічних систем загального призначення.

Поточні системи виявлення замінюють класифікатори для здійснення виявлення. Для виявлення об'єкта ці системи беруть класифікатор для цього об'єкта і оцінюють його в різних місцях і масштабах на тестовому зображенні. Такі системи, як моделі деформованих деталей (Deformable Parts Model - DPM [32]), використовують підсувний віконний підхід, коли класифікатор працює в рівномірно розташованих місцях по всьому зображенню.

Більш сучасні підходи, такі як R-CNN, використовують методи пропозицій регіону, щоб спершу генерувати потенційні обмежувальні поля на зображенні, а потім запустити класифікатор для цих запропонованих вікон. Після класифікації післяобробка використовується для уточнення обмежувальних прямокутників, усунення дублікатів виявлення та повторного перегляду полів на основі інших об'єктів на сцені. Ця складна послідовність дій повільна та важка для оптимізації, оскільки кожен окремий компонент повинен тренуватися окремо.

Автори розглянутого підхіда переосмислюють виявлення об'єктів як єдину проблему регресії, прямо від пікселів зображення до обмежувальних координат прямокутника та ймовірностей класу. Використовуючи їх систему,

людина дивиться лише один раз (YOLO) на зображення, щоб передбачити, які об’єкти є і де вони знаходяться.

Ідея YOLO дуже проста, її ілюструє рис. 3.1. Єдина згорткова мережа одночасно прогнозує кілька обмежувальних прямокутників та ймовірності класів для цих регіонів. YOLO тренується на повних зображеннях і безпосередньо оптимізує продуктивність виявлення. Ця уніфікована модель має ряд переваг перед традиційними методами виявлення об'єктів.

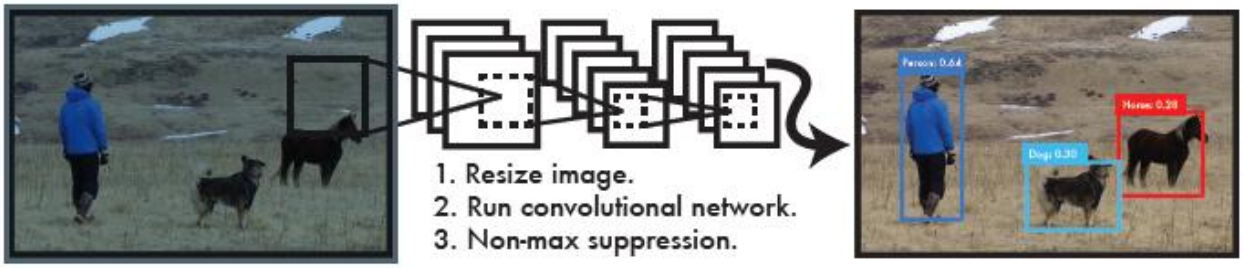


Рисунок 3.1 — Система виявлення YOLO

По-перше, YOLO – надзвичайно швидка система. Оскільки кадр розглядається як проблема регресії, то не потрібен складний конвеєр. Нейронна мережа запускається на нове зображення в тестовий час, щоб передбачити виявлення. Базова мережа працює зі швидкістю 45 кадрів в секунду без пакетної обробки на GPU Titan X, а швидка версія працює з більш ніж 150 кадрів в секунду. Це означає, що можливо обробляти потокове відео в режимі реального часу менше 25 мілісекунд затримки. Крім того, YOLO досягала більш ніж удвічі середньої точності порівняно з іншими системами реального часу на той час.

По-друге, YOLO глобально міркує про зображення під час прогнозування. На відміну від методів, що базуються на розсувних вікнах та регіонах, YOLO бачить усе зображення під час тренувань та тестування, тому воно неявно кодує контекстну інформацію про класи, а також їх зовнішній

вигляд. Fast R-CNN [36], топовий метод виявлення, помиляється на фонових зображеннях виявляючи об'єкти, оскільки він не бачить більшого контексту. YOLO робить менше половини кількості помилок фону порівняно з Fast R-CNN.

По-третє, YOLO засвоює узагальнюючі уявлення предметів. Під час навчання природним зображенням та тестування на художніх творах YOLO перевершує найкращі методи виявлення, такі як DPM та R-CNN з великим запасом. Оскільки YOLO є дуже узагальнювальним, то менше шансів вийти з ладу при застосуванні до нових доменів або несподіваних входів.

YOLO як і раніше відстає від сучасних систем виявлення. Хоча він може швидко ідентифікувати об'єкти на зображеннях, він не може точно локалізувати деякі об'єкти, особливо маленькі. Ці компроміси будуть розглянуті далі в експериментах, проведених авторами YOLO.

3.1 Уніфіковане виявлення

В YOLO об’єднуються окремі компоненти виявлення об'єктів в єдину нейронну мережу. Мережа використовує особливості всього зображення для прогнозування кожного обмежувального прямокутника. Він також передбачає всі обмежувальні прямокутники для всіх класів на зображенні одночасно. Це означає, що мережа міркує глобально про повне зображення та всі об'єкти на зображенні. Конструкція YOLO дозволяє здійснювати повний цикл навчання зі швидкістю в режимі реального часу, зберігаючи високу середню точність.

Розглянута система ділить вхідне зображення на сітку $S \cdot S$. Якщо центр об'єкта потрапляє в клітину сітки, ця клітина відповідає за виявлення цього об'єкта.

Кожна клітина сітки прогнозує B обмежуючих прямокутників та оцінку достовірності для цих прямокутників. Ці показники достовірності відображають наскільки впевненою є модель у тому, що обмежувальний

прямокутник містить предмет, а також наскільки точною вона вважає обмежувальний прямокутник, що вона прогнозує. Формально впевненість визначається як *Pr(Object)*IOU(truth, pred)*, де *Pr(Object)* — ймовірність того, що клітина сітки містить об’єкт, а *IOU(truth, pred)* — значення метрики близькості між прогнозованим та істинним обмежувальним полем. Якщо в цій клітині немає жодного об'єкта, показник достовірності повинен бути нульовим. В іншому випадку показник достовірності повинен дорівнювати перетину між союзом (*IOU*) між передбачуваним обмежувальним прямокутником і реальним.

Кожний обмежувальне поле складається з 5 прогнозів: *x, y, w, h* та впевненості. Координати *(x; y)* являють собою центр вікна відносно меж клітини сітки. Ширина та висота прогнозуються відносно всього зображення. Нарешті, прогноз достовірності являє собою *IOU* між передбачуваним полем та будь-яким реальним.

Кожна клітина сітки також прогнозує ймовірності С умовного класу, $\Pr(Class_i \mid Object)$. Ці ймовірності обумовлені в клітині сітки, що містить об'єкт. В першій версії YOLO прогнозується лише один набір ймовірностей класу на клітині сітки, незалежно від кількості обмежувальних прямокутників *B*.

У тестовий час ймовірності умовного класу та індивідуальні прогнози достовірності вікна множаться:

$$\Pr(Class_i \mid Object) * \Pr(Object) * IOU_{pred}^{truth} = \Pr(Class_i) * IOU_{pred}^{truth}, \quad (3.1)$$

що дає оцінку достовірності для кожного поля. Ці показники кодують як ймовірність появи цього класу у обмежувальному полі, так і наскільки добре передбачуване поле відповідає об'єкту. На рис. 3.2 представлений цикл роботи нейронної мережі.

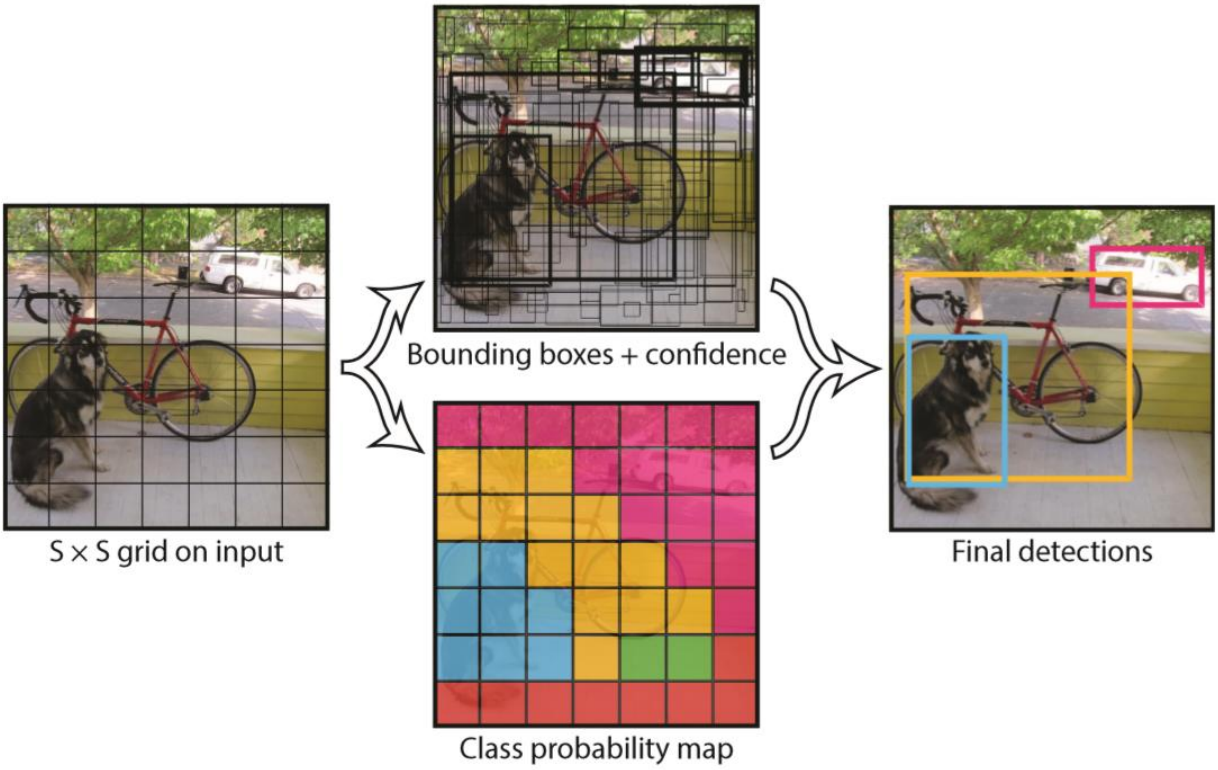


Рисунок 3.2 — Схема роботи нейронної мережі YOLO

3.2 Архітектура мережі YOLO

Модель реалізована як звивиста нейронна мережа та оцінена на базі даних детектування PASCAL VOC [49]. Початкові згорткові шари мережі витягують з зображення особливості, тоді як повністю пов'язані шари прогнозують вихідні ймовірності та координати.

Мережева архітектура, що розглядається, натхненна моделлю GoogLeNet [50] для класифікації зображень. У цій мережі є 24 згорткових шари, а потім 2 повністю з'єднаних шари. Замість початкових модулів, які використовує GoogLeNet, автори YOLO просто використовують редукційні шари 1×1 з подальшими 3×3 згортковими шарами, подібними до Lin et al. [33]. Повна архітектура мережі YOLO показана на рис. 3.3.

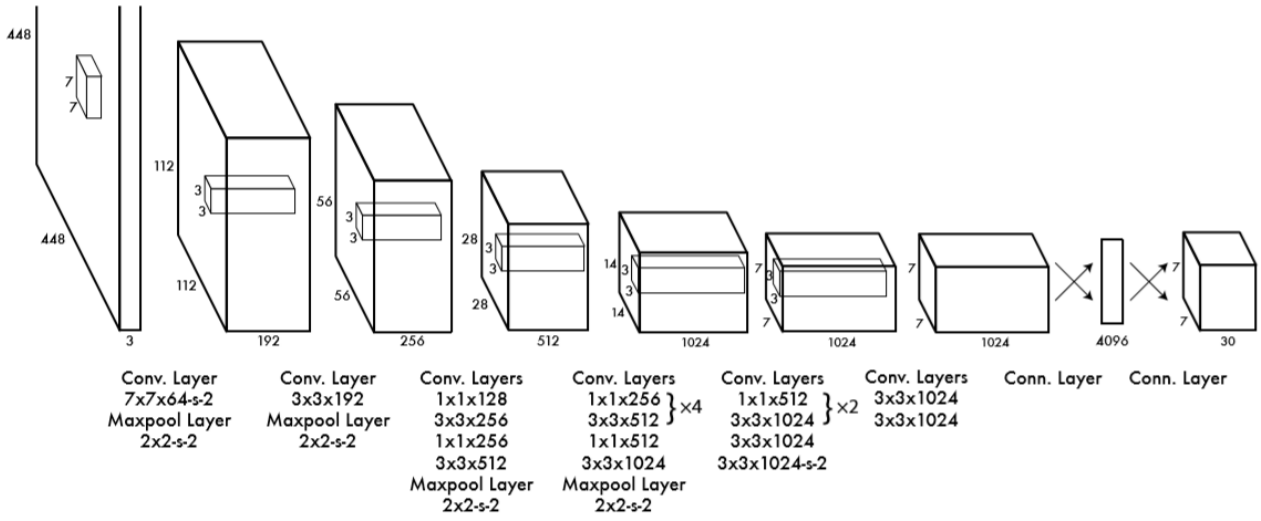


Рисунок 3.3 — Архітектура першої версії YOLO

Автори також зробили швидку версію YOLO, розроблену для швидкого виявлення об'єктів. Швидка YOLO використовує нейронну мережу з меншою кількістю згорткових шарів (9 замість 24) та меншою кількістю фільтрів у цих шарах. Крім розміру мережі, всі параметри тренування та тестування однакові між YOLO та Fast YOLO.

Вихідний вихід нашої мережі — це тензор прогнозів 7×7×30.

3.3 Навчання YOLO

Автори попередньо навчають звивисті шари в наборі даних змагань 1000-класу ImageNet [51]. Для попереднього навчання використовують перші 20 згорткових шарів з рис. 3.3, потім шар середнього об'єднання та повністю пов'язаний шар. Автори тренують цю мережу приблизно тиждень і досягають точності топ-5 в 88% на наборі валідації ImageNet 2012, порівнянної з моделями GoogLeNet в модельному зоопарку Caffe [52]. Darknet використовується для всіх тренувань і висновків [53].

Потім модель для класифікації була перетворена на модель для виявлення. Ren et al. показують, що додавання як звивистих, так і з'єднаних

шарів до попередньо преднавчених мереж може підвищити продуктивність [54]. Слідуючи їх прикладу, авторами було додано чотири згорткових шари та два повністю пов'язані шари з випадково ініціалізованими вагами. Для виявлення часто потрібна чітка візуальна інформація, тому була збільшена вхідна роздільна здатність мережі з 224×224 до 448×448.

Остаточний шар YOLO прогнозує як ймовірності класу, так і координати обмежувального прямокутника. Автори нормалізують ширину та висоту обмежувального поля за шириною та висотою зображення, щоб вони падали між 0 та 1. Автори також параметризують *x* та *y* координати обмежувального прямокутника, щоб бути зрушеннями для певного місця клітини сітки, щоб вони також були обмежені між 0 та 1.

В YOLO використовують функцію лінійної активації для фінального шару, а всі інші шари використовують наступну герметичну випрямлену лінійну активацію:

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \tag{3.2}$$

У якості критерія оптимізації авторами було обрано суму квадратів помилок на виході моделі. Цей критерій був обраний, тому що його легко оптимізувати, проте він не ідеально відповідає меті досягнення максимальної середньої точності. Він зважує помилку локалізації в рівній мірі з помилкою класифікації, яка може бути не ідеальною. Також у кожному зображенні багато клітин сітки які не містять жодного об'єкта. Це підштовхує «довіреність» балів цих клітин до нуля, часто перекриваючи градієнт від клітин, які містять об'єкти. Це може призвести до нестабільності моделі, що призведе до того, що навчання рано розходиться.

Щоб виправити це, автори збільшують втрати від прогнозування координат обмежувального прямокутника та зменшують втрати від прогнозування впевненості для полів, у яких немає об’єктів. Для цього використовують два параметри — λ_{coord} і λ_{noobj}. Встановлюють λ_{coord} = 5 і λ_{noobj} =0.5.

Сума квадратів помилок також однаково зважує помилки у великих обмежувальних прямокутників та невеликих. Показник помилок повинен свідчити про те, що малі відхилення у великих обмежувальних прямокутників мають значення менше, ніж у малих. Щоб частково вирішити це, автори передбачають квадратний корінь ширини та висоти обмежувального поля замість ширини та висоти безпосередньо.

YOLO прогнозує кілька обмежувальних прямокутників на клітину сітки. Під час тренування хочеться, щоб за кожний об'єкт відповідав лише один передбачувач обмежувальної коробки. Автори призначають один детектор «відповідальним» за прогнозування об'єкта, на основі якого прогноз має найвищий поточний *IOU* з істинним. Це призводить до спеціалізації між передбачувачами обмежувальної коробки. Кожен передбачувач навчається краще прогнозувати певні розміри, співвідношення сторін або класів об'єктів, покращуючи загальну картину.

Під час тренінгу оптимізується функція витрат:

$$
\begin{aligned}
&\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(x_i - \hat{x}_i \right)^2 + \left(y_i - \hat{y}_i \right)^2 \right] \\
&+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned}
$$

(3.3)

де 1_i^{obj} позначає, якщо об'єкт з'являється в клітині і та 1_{ij}^{obj} означає, що передбачувач j-го обмежувального поля в клітині і "відповідає" за це прогнозування.

Слід зауважити, що функція витрат карає лише помилку класифікації, якщо в цій клітині сітки присутній об'єкт (отже, це – ймовірність умовного класу, обговорена раніше). Він також карає помилку координати обмежувальної коробки, лише якщо цей предиктор несе відповідальність за істинну коробку (тобто має найвищий *IOU* з будь-якого передбачувача цієї клітини сітки).

Автори навчали мережу близько 135 епох на наборах даних по навчанню та валідації з PASCAL VOC 2007 та 2012. Під час тестування на PASCAL VOC 2012 року також включають дані тесту VOC 2007 для тренування. Протягом тренування використовується розмір батча 64, імпульс 0.9 і розпад 0.0005.

Для перших епох повільно підвищували швидкість навчання η з 0.001 до 0.01. Якщо починати з високої швидкості навчання, модель часто

розходиться через нестабільні градієнти. Навчання продовжувалося з ŋ =0.01 для 75 епох, потім ŋ =0.001 для 30 епох і остаточно ŋ =0.0001 протягом 30 епох.

Щоб уникнути перенавчання, рекомендується використовувати dropout та широке доповнення даних. Слой dropout, що випадає з вірогідністю = 0.5 після першого з'єднаного шару, запобігає спільній адаптації між шарами [55]. Для збільшення даних вводиться випадкове масштабування та переклади до 20% від вихідного розміру зображення. Також випадково регулюється експозиція та насиченість зображення до коефіцієнта 1,5 в кольоровому просторі HSV.

Як і в тренуванні, прогнозування результатів для тестового зображення вимагає лише однієї мережевої оцінки. На PASCAL VOC мережа прогнозує 98 обмежувальних коробок на зображенні та ймовірності класу для кожного вікна. YOLO надзвичайно швидка у тестовий час, оскільки вимагає лише єдиної оцінки мережі, на відміну від класифікованих методів.

Дизайн мережі забезпечує просторове розмаїття в передбачувальних вікнах. Часто зрозуміло, до якої клітини сітки потрапляє об’єкт, і мережа передбачає лише одне поле для кожного об’єкта. Однак деякі великі об'єкти або об'єкти біля межі кількох клітин можуть бути добре локалізовані кількома клітинами. Non-maximal suppression можна використовувати для фіксації цих кількох виявлень. Незважаючи на ефективність роботи, як це стосується R-CNN або DPM, Non-maximal suppression додає 23% в mAP.

3.4 Обмеження YOLO

YOLO накладає сильні просторові обмеження для прогнозування обмежувальної коробки, оскільки кожна комірка сітки передбачає лише два поля і може мати лише один клас. Це просторове обмеження обмежує кількість

об'єктів поблизу, які може передбачити модель. Авторська модель бореться з дрібними предметами, які з'являються в групах, наприклад, птахи.

Оскільки модель вчиться прогнозувати обмежувальні поля з даних, вона намагається узагальнити об'єкти в нових або незвичайних співвідношеннях сторін або конфігураціях. Модель також використовує відносно грубі функції для прогнозування обмежувальних коробок, оскільки в архітектурі є кілька шарів зменшення розміру з вхідного зображення.

Нарешті, поки тренування йде на функції втрат, яка наближає ефективність виявлення, авторська функція втрат трактує однакові помилки у малих обмежувальних полях порівняно з великими обмежувальними полями. Невелика помилка у великій коробці, як правило, є доброякісною, але невелика помилка у маленькій коробці має набагато більший вплив на IOU. Тож головне джерело помилок — неправильна локалізація.

4 КОМБІНОВАНИЙ НЕЙРОННО-МЕРЕЖЕВИЙ ПІДХІД ДЛЯ ВИЯВЛЕННЯ ТЕКСТУ

Виявлення тексту в природних зображеннях є досить складною задачею, тож загальні підходи у виявленні об’єктів можуть не працювати достатньо ефективно, оскільки текст окрім візуальних характеристик також має семантичні. Цей фактор призводить до того, що нейронна мережа може виявляти об’єкти, які схожі на текст, але текстом не являються. Приклад такого можна побачити на рис. 4.1 — 4.2.

Рисунок 4.1 — Об’єкт, схожий на текст

На першому рисунку нейронна мережа може сприйняти геометричні фігури за текст.

Рисунок 4.2 — Об’єкт, схожий на текст

На другому рисунку відомий усім логотип мессенджера Telegram може бути сприйнятий як за цифру 7.

Практика показує, що таких прикладів безліч, що в свою чергу ускладнює ефективне виявлення тексту з природних зображень, оскільки неможливо підготувати нейронну мережу до всіх можливих варіантів, якщо немає інструменту глибокої генералізації на рівні архітектури.

Також відомо, що згорткові слої в процесі тренування вивчають в більшості візуальні характеристики об’єктів, ніж семантичну. Це призводить до того, що страждає генералізація згорткових нейронних мереж у цілому. Яскравий тому приклад — феномен адверсаріал атаки [56]. На рис. 4.3 наведено приклад цього явища.

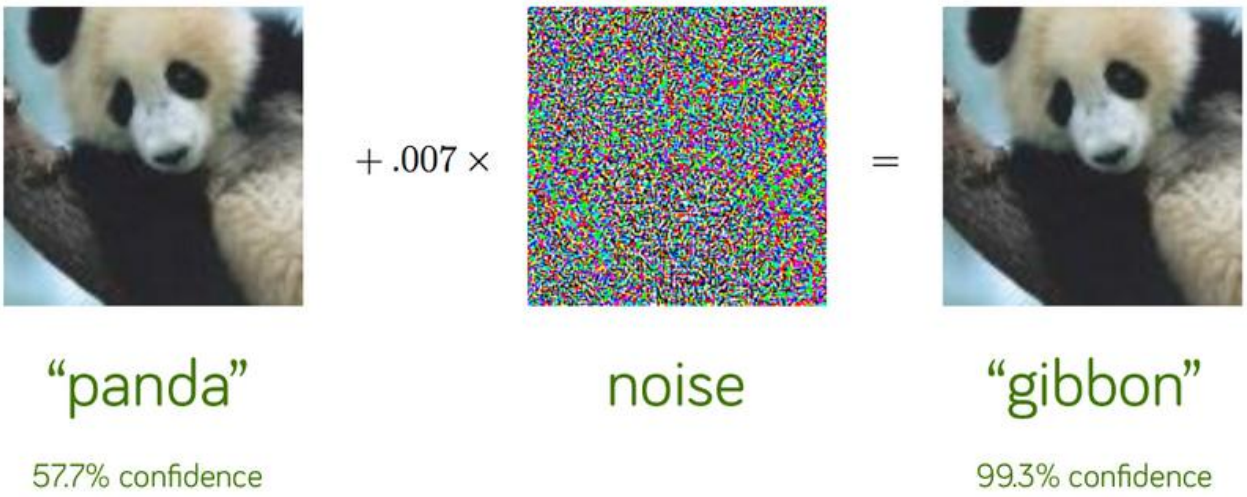


Рисунок 4.3 — Приклад адверсаріал атаки

Ці фактори наводять на те, що нейронну мережу потрібно схиляти до вивчення семантичної інформації в процесі навчання в будь-якій задачі, а особливо в задачі виявлення тексту в природних зображеннях.

4.1. Побудова архітектури глибинної мережі для виявлення тексту в реальних сценах

Як видно з попередніх досліджень, CRAFT є дуже ефективним підходом у виявленні символів. Навчання нейронної мережі виявляти теплові карти тексту у природних зображеннях, де максимальна інтенсивність вказує на вірогідність того, що піксель є центром символу позитивно сприяє на генералізацію та результат. Але все ж таки текст складається зі слів, а не символів, тому розглянутий підхід потребує додаткового кроку, мета якого — групування виявлених символів у слова.

Такий крок є алгоритмічним, тобто таким, який не навчається і має декілька гіперпараметрів, які треба ще підібрати. Крім того, навіть добре підібрані параметри на одних даних можуть показувати високу ефективність, в той час як на інших — низьку.

З іншого боку, архітектуру YOLO можна використовувати для виявлення тексту на рівні слів, але результати таких намагань не є досконалими, оскільки YOLO — це фреймворк для виявлення загальних об’єктів, які зустрічаються в наборі даних ImageNet. Архітектура YOLO не має модулів, які б допомогли ефективно виявляти текст.

Виходячи з перерахованих особливостей обох підходів, у цій роботі пропонується об’єднати архітектури YOLO та CRAFT в єдину нейронну мережу. Новий підхід виключає недоліки підходу CRAFT — тепер немає потреби в будь-якому алгоритмі після обробки з одного боку, а з іншого — нова нейронна мережа отримає усі переваги від архітектури YOLO — ефективного та швидкого детектора тексту.

4.2 Функція втрат

Комбінований підхід навчається не тільки виявляти обмежувальні поля слів, для чого використовуються функції втрат впевненості нейронної мережі у тому, що клітина сітки містить об’єкт, його не містить, а також функції втрат, які сприяли б регресії обмежувальних полів, як то робиться в підході YOLO. Також потрібно навчати нейронну мережу виявляти теплові карти символів та їх взаємозв’язки, як то робиться в підході CRAFT.

4.2.1 Регресія обмежувальних полів.

Окремо розглянемо регресію обмежувальних полів в комбінованому підході, оскільки відповідні функції втрат, які використовуються в YOLO на сьогоднішній час, є малоефективними та застарілими.

Для ефективного виявлення обмежувальних полів потрібна функція втрат, яка б сприяла не тільки навчанню безпосередньо виявляти координати поля, але ще б направляла нейронну мережу вчити геометричні особливості обмежувального прямокутника.

Цим вимогам повністю задовольняє Distance Intersection Over Union Loss [57]. Розглянемо цей підхід.

Чи можливо безпосередньо мінімізувати нормовану відстань між передбачуваним полем і цільовим полем для досягнення більш швидкої сходимості?

Як правило, функції втрат на базі IOU метрики можуть бути визначені як

$$L = 1 - IOU + R(B, B^{gt}), \tag{4.1}$$

де $R(B, B^{gt})$ — штраф для прогнозованого поля B та цільового поля B^{gt}. Створюючи належні штрафні строки, далі буде розглянута DIoU функція втрат, щоб відповісти на вищезазначене питання.

Щоб відповісти на запитання, автори [57] пропонують мінімізувати нормовану відстань між центральними точками двох обмежувальних коробок, а штрафний строк визначити як

$$R_{DIoU} = \frac{\rho^2\left(b, b^{gt}\right)}{c^2}, \tag{4.2}$$

де b і b^{gt} позначають центральні точки обмежувальних полів B і B^{gt}, ρ — евклідова відстань, а c — діагональна довжина найменшого обмежувального поля, що охоплює дві коробки. І тоді функцію втрати DIoU можна визначити як

$$L_{DIoU} 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2}, \tag{4.3}$$

Як показано на рис. 4.4, штрафний строк втрати DIoU безпосередньо мінімізує відстань між двома центральними точками

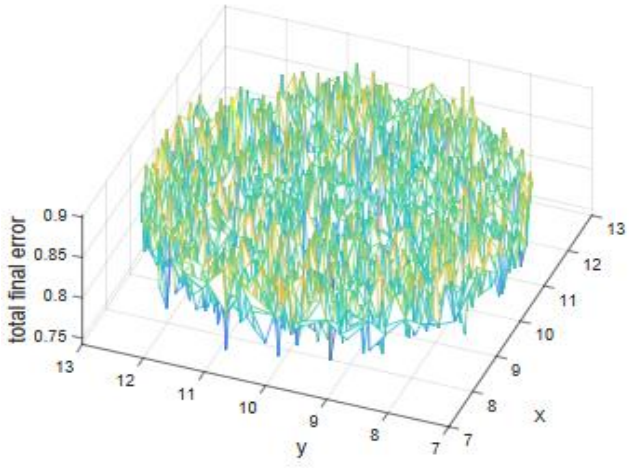


Рисунок 4.4 — Візуалізація тренування мережі з функцією втрат DIoU.

4.2.2 Прогнозування обмежувального поля

Слідом за YOLOv3 [58] комбінована система прогнозує обмежувальні коробки, використовуючи кластери розмірів як анкори Faster r-cnn. Мережа прогнозує 4 координати для кожного обмежувального вікна, t_x, t_y — (координати центру по осям x та y) t_w, t_h — (прогнозоване зміщення ширини та висоти). Якщо клітина зміщена у верхньому лівому куті зображення на (c_x, c_y), а анкор має ширину p_w та висоту p_h, то прогнози відповідають наступним формулам:

$$
\begin{aligned}
b_x &= \sigma(t_x) + c_x \\
b_y &= \sigma(t_y) + c_y \\
b_w &= p_w e^{t_w} \\
b_h &= p_h e^{t_h}
\end{aligned}
\tag{4.4}
$$

На рис. 4.5 показан зв’язок між сіткою клітин та обчисленням прогнозованих координат обмежувальних полей слів.

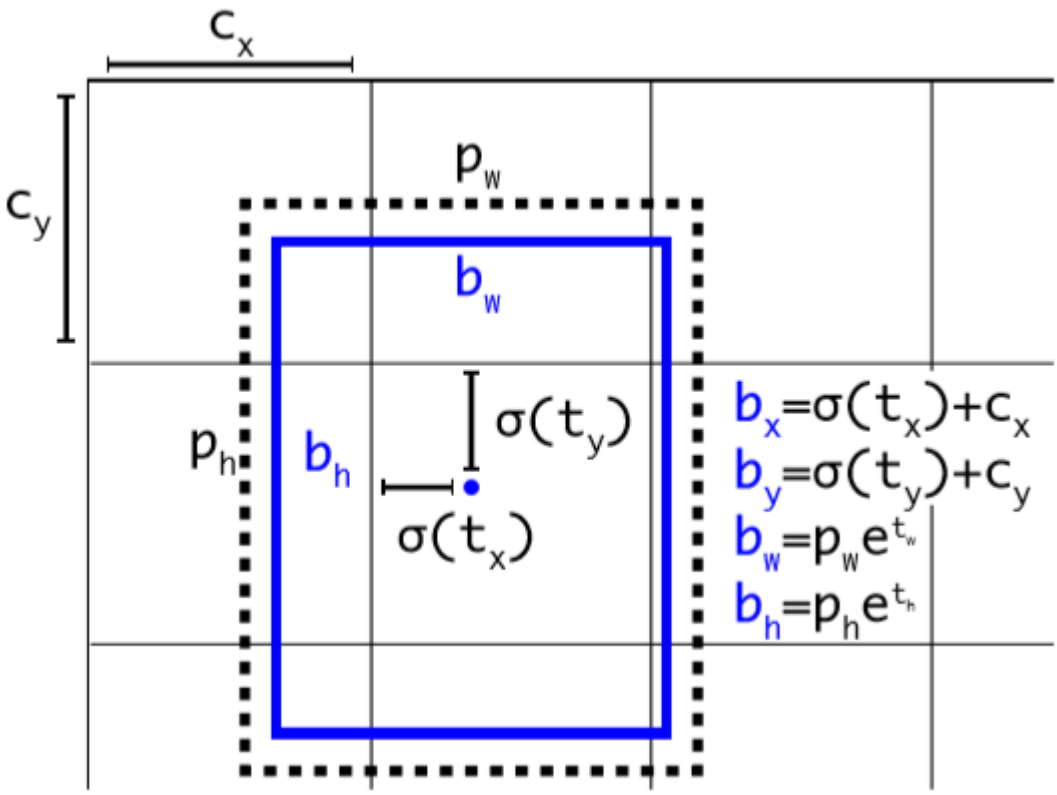


Рисунок 4.5 — Зв’язок між сіткою клітин та обчисленням прогнозованих координат обмежувальних полей слів.

4.3 Комбінація архітектур

Основою комбінованої архітектури буде мережа CRAFT. З останнього блоку CRAFT, який прогнозує теплові карти мапи ознак переходять в основну частину YOLO замість оригінального зображення, як то роблять автори YOLO. Тобто вилучувач ознак, взятий з архітектури YOLO, на вхід отримає мапу ознак, з якої робляться теплові карти символів та їх зв’язків. Подальша схема обчислень комбінованою нейронною мережею ідентична YOLO.

5 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ КОМБІНОВАНОГО ПІДХОДУ ДО ВИЯВЛЕННЯ ОБЛАСТЕЙ ТЕКСТУ В РЕАЛЬНИХ СЦЕНАХ

5.1 Набори даних

ICDAR2013 (IC13) був випущений під час конкурсу на міцне читання ICDAR 2013 для фокусированного виявлення тексту сцени, що складається з зображень високої роздільної здатності, 229 для тренувань та 233 для тестування, що містять тексти англійською мовою. Анотації знаходяться на рівні слова, використовуючи прямокутні поля.

ICDAR2015 (IC15) був представлений у змаганнях з читання читання текстів сцени ICDAR 2015 для виявлення випадкових текстів сцени, що складається з 1000 навчальних зображень та 500 зображень для тестування, обидва з текстами англійською мовою. Анотації знаходяться на рівні слова, використовуючи чотирикутні поля.

ICDAR2017 (IC17) містить 7200 навчальних зображень, 1800 зображень валідації та 9000 тестових зображень з текстами 9 мовами для багатомовного виявлення тексту сцени. Подібно до IC15, текстові області в IC17 також позначаються 4 вершинами чотирикутників.

MSRA-TD500 (TD500) містить 500 природних зображень, які розділені на 300 навчальних зображень та 200 тестових зображень, зібраних як у приміщенні, так і на вулиці за допомогою кишенькової камери. Зображення містять англійський та китайський сценарії. Текстові області позначаються обернеными прямокутниками.

TotalText (TotalText), нещодавно представлений в ICDAR 2017, містить 1255 навчальних та 300 тестових зображень. Він особливо надає вигнуті тексти, які позначаються багатокутниками та транскрипціями на рівні слів.

CTW-1500 (CTW) складається з 1000 навчальних та 500 тестових зображень. Кожне зображення має вигнуті текстові екземпляри, які позначаються багатокутниками з 14 вершинами.

Процедура навчання включає два етапи: автори вперше використовують набір даних SynthText [6] для навчання мережі для 50-ти ітераційних дій, після чого кожен набір даних орієнтирів приймається для того, щоб налагодити модель. Деякі області тексту в наборах даних ICDAR 2015 та ICDAR 2017 ігноруються в навчанні. Для тренувань з декількома GPU, GPU для тренування та спостереження відокремлюються, а псевдо-GT, що генеруються GPU для контролю, зберігаються в пам'яті. Під час налагодження також використовується набір даних SynthText зі швидкістю 1:5, щоб гарантувати, що області символів точно відокремлені. Для виведення текстур, подібних до текстів у природних сценах, застосовується онлайн жорсткий негативний видобуток (Online Hard Example Mining - HEM [31]) у співвідношенні 1:3. Також застосовуються основні методи збільшення даних, такі як кропи, повороти та / або кольорові варіації.

Для навчання під слабким наглядом потрібні два типи даних; чотирикутні анотації для обрізання зображень слова та транскрипції для обчислення довжини слова. Набори даних, що відповідають цим умовам, — це IC13, IC15 та IC17. Інші набори даних, такі як MSRA-TD500, TotalText і CTW-1500, не відповідають вимогам. MSRA-TD500 не забезпечує транскрипції, тоді як TotalText і CTW-1500 надають лише полігонові анотації. Тому автори тренували CRAFT лише на наборах даних ICDAR, а на інших перевіряли без налагодження. Дві різні моделі навчаються наборам даних ICDAR. Перша модель навчається на IC15 тільки для оцінки IC15. Друга модель навчається як на IC13, так і на IC17 разом, яка використовується для оцінки інших наборів даних. Жодні додаткові зображення не використовуються для тренувань. Кількість ітерацій для налаштування встановлено на 25 тисяч.

Це стосовно текстових наборів даних. YOLO, на відміну від CRAFT являється детектором загальних об’єктів, які зустрічаються в наборі даних ImageNet. Тож на рівні архітектури YOLO не має жодних модулей, які б допомогали ефективно виявляти текст. Тож в цій роботі будуть досліджені авторські тренування на загальних наборах даних.

Тепер стосовно генералізації нейронної мережі. Академічні набори даних для виявлення об'єктів черпають дані навчання та тестування з одного розподілу. У реальних ситуаціях важко передбачити всі можливі випадки використання, і дані тесту можуть відрізнятися від того, що бачила система раніше [59]. Тому в експериментах з YOLO також використовували набори даних Picasso [60] та People-Art [61] – два набори для тестування виявлення людей на художньому творі.

5.2 Результати експериментів з мережею CRAFT

Набори даних з анотаціями на рівні чотирикутника (ICDAR та MSRA-TD500). Усі експерименти проводяться з єдиною роздільною здатністю зображення. Довга сторона зображень у IC13, IC15, IC17 та MSRA-TD500 змінюється відповідно до розмірів 960, 2240, 2560 та 1600 відповідно. У Таблиці 1.1 наведено середні показники різних методів на наборів даних ICDAR та MSRA-TD500. Для справедливого порівняння з методами в кінці, автори включають їх результати лише для виявлення, посилаючись на оригінальні статті. В результаті досягненні найсучасніші показники на всіх наборах даних. Крім того, CRAFT працює на 8,6 FPS на базі даних IC13, що порівняно швидко, завдяки простої, але ефективної післяобробки.

Для MSRA-TD500 анотації надаються на рівні рядка, включаючи пробіли між словами у полі. Тому застосовується крок після обробки для комбінування текстових полів. Якщо права частина однієї коробки та ліва частина іншої розташовані досить близько, два вікна поєднуються між собою.

Незважаючи на те, що налаштування не проводиться на навчальному наборі TD500, CRAFT перевершує всі інші методи, як показано в табл. 5.1.

Набори полігонних наборів даних (TotalText, CTW-1500). Складно тренувати модель на TotalText та CTW1500, оскільки їх анотації мають полігональну форму, що ускладнює обрізання області тексту для розбиття полів символів під час тренування під слабким контролем. Отже, автори використовували лише навчальні зображення з IC13 та IC17, і налагодження не проводилось для вивчення навчальних зображень, наданих цими наборами даних. На етапі висновку була використана післяобробка багатокутника після обробки з оцінки регіону, щоб впоратися з наданими анотаціями полігонального типу.

Таблиця 5.1 — Показники ефективності моделей на різних наборах даних

Method	IC13(DetEval)			IC15			IC17			MSRA-TD500			FPS
	R	P	H	R	P	H	R	P	H	R	P	H	
Zhang et al.	78	88	83	43	71	54	-	-	-	67	83	74	0.48
Yao et al.	80.2	88.8	84.3	58.7	72.3	64.8	-	-	-	75.3	76.5	75.9	1.61
SegLink	83.0	87.7	85.3	76.8	73.1	75.0	-	-	-	70	86	77	20.6
SSTD	86	89	88	73	80	77	-	-	-	-	-	-	7.7
Wordsup	87.5	93.3	90.3	77.0	79.3	78.2	-	-	-	-	-	-	1.9
EAST	-	-	-	78.3	83.3	80.7	-	-	-	67.4	87.3	76.1	13.2
He et al	81	92	86	80	82	81	-	-	-	70	77	74	1.1
R2CNN	82.6	93.6	87.7	79.7	85.6	82.5	-	-	-	-	-	-	0.4
TextSnake	-	-	-	80.4	84.9	82.6	-	-	-	73.9	83.2	78.3	1.1
TextBoxes++	86	92	86	78.5	87.8	82.9	-	-	-	-	-	-	2.3
EAA	87	88	88	83	84	83	-	-	-	-	-	-	-
Mask TextSpotter	88.1	94.1	91.0	81.2	85.8	83.4	-	-	-	-	-	-	4.8
PixelLink	87.5	88.6	88.1	82.0	85.5	83.7	-	-	-	73.2	83.0	77.8	3.0
RRD	86	92	89	90.0	88.0	84.8	-	-	-	73	87	79	10
Lyu et al.	84.4	92.0	88.0	79.7	89.5	84.3	70.6	74.3	72.4	76.2	87.6	81.5	5.7
FOTS	-	-	87.3	82.0	88.8	85.3	57.5	79.5	66.7	-	-	-	23.9
CRAFT	93.1	97.4	95.2	84.3	89.8	86.9	68.2	80.6	73.9	78.2	88.2	82.9	8.6

Експерименти для цих наборів даних також виконуються з однією роздільною здатністю зображення. Більш довгі сторони зображень у наборах TotalText та CTW-1500 змінено до 1280 та 1024 відповідно. Результати експериментів для наборів даних полігонових типів наведені в табл. 5.2. Здатність CRAFT до локалізації окремих символів дозволяє досягти більш надійної та чудової продуктивності у виявленні довільно сформованих текстів порівняно з іншими методами. Зокрема, набір даних TotalText має різноманітні деформації, включаючи вигнуті тексти, як показано на рис. 5.1, для яких адекватний висновок детекторів тексту на основі чотирикутників неможливий. Тому дуже обмежена кількість методів може бути оцінена на цих наборах даних.

Рисунок 5.1 — Приклад вигнутих текстів в наборі даних TotalText.

Таблиця 5.2 — Порівняння Craft мережі та інших на полігональних наборах даних

Method	TotalText			CTW-1500		
	R	P	H	R	P	H
CTD+TLOC	-	-	-	69.8	77.4	73.4
MaskSpotter	55.0	69.0	61.3	-	-	-
TextSnale	74.5	82.7	78.4	85.4	67.9	75.6
CRAFT	79.9	87.6	83.6	81.1	86.0	83.5

У випадку набору даних CTW-1500 співіснують дві складні характеристики, а саме анотації, які надаються на рівні лінії та мають довільні багатокутники. Щоб допомогти CRAFT у таких випадках, разом із CRAFT використовується невелика мережа покращення зв'язку (LinkRefiner). Вхідним кодом покращувача зв’язків є конкатенація балів регіону, показник ефективності та проміжна карта ознак CRAFT, а вихід — це відповідна оцінка ефективності, скоригована для довгих текстів (рис. 5.2).

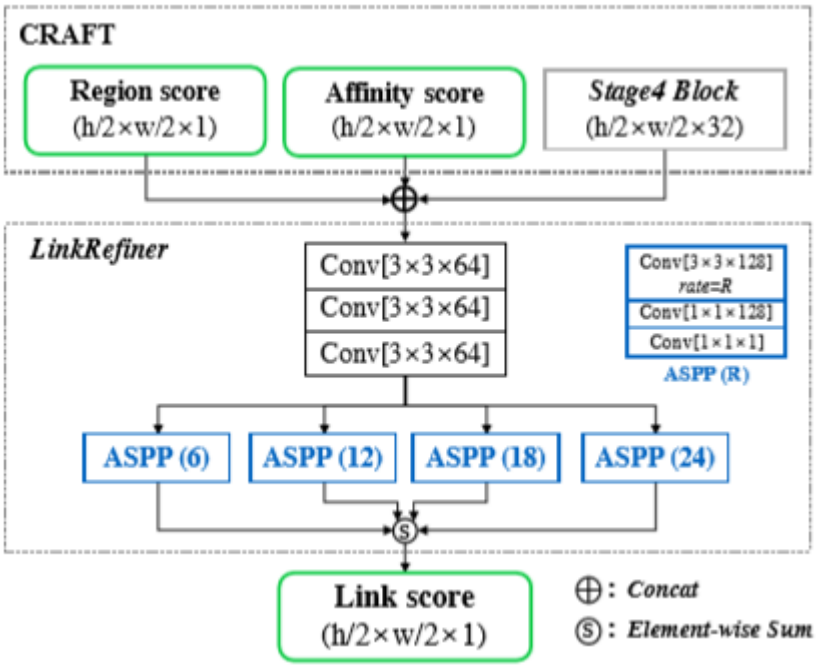


Рисунок 5.2 — Архітектура покращувача зв’язків

Для комбінування символів використовується покращена оцінка близькості замість оригінальної оцінки близькості, тоді генерація багатокутника виконується так само, як і для TotalText. На даних CTW-1500 під час заморожування CRAFT навчається лише LinkRefiner. Детальна реалізація LinkRefiner розглянута у додаткових матеріалах. Як показано в табл. 5.2, запропонований спосіб досягає найсучасніших показників.

Все це були одномасштабні експерименти на всіх наборах даних, хоча розміри текстів дуже різноманітні. Це відрізняється від більшості інших методів, які покладаються на багатомасштабні тести для вирішення проблеми дисперсії масштабу. Ця перевага виходить із властивості авторського методу локалізації окремих символів, а не всього тексту. Порівняно невелике сприйнятливе поле достатньо для покриття одного символу у великому зображенні, що робить CRAFT надійним у виявленні текстів різного масштабу.

Також, як показано в табл. 5.3, з аналізу випадків відмов очікується, що CRAFT модель отримує користь від результатів розпізнавання, особливо коли основні слова правди розділені семантикою, а не візуально.

Таблиця 5.3 — Порівняння CRAFT та інших “від початку до кінця” методами

Method	IC13	IC15	IC17
CTD+TLOC	91.7	86.9	-
MaskSpotter	90	87	-
TextSnale	92.8	89.8	70.8
CRAFT	95.2	86.9	73.9

Метод CRAFT демонструє високу точність у трьох різних наборах даних без додаткової настройки. Це свідчить про те, що модель здатна фіксувати загальні характеристики текстів, а не переоцінювати певний набір даних.

5.3 Результати експериментів з мережею YOLO

Спочатку буде наведемо порівняння YOLO з іншими системами виявлення в реальному часі на PASCAL VOC 2007. Щоб зрозуміти відмінності між варіантами YOLO та R-CNN, авторами YOLO досліджуються помилки VOC 2007, зроблені YOLO та Fast R-CNN, однією з найбільш

високопродуктивних версій R-CNN [11]. На основі різних профілів помилок видно, що YOLO можна використовувати для перегляду швидких виявлень R-CNN та зменшення помилок із фонових помилкових позитивів, що дає значне підвищення продуктивності. Далі також представлені результати VOC 2012 та порівняно mAP з сучасними методами. Нарешті, становиться видно, що YOLO узагальнює нові домени краще, ніж інші детектори на двох наборах даних художнього твору.

Багато дослідницьких зусиль з виявлення об'єктів зосереджуються на швидкому зростанні стандартних процесів виявлення [36-40]. Однак, лише Sadeghi та ін. насправді пропонують систему виявлення, яка працює в режимі реального часу (30 кадрів в секунду або краще) [39]. Автори підходу порівнюють YOLO з їх GPU-реалізацією DPM, яка працює або на 30 Гц, або на 100 ГГц. Хоча підходи інших авторів не є настільки швидкими, щоб прогнозувати результати в режимі реального часу, тут все одно буде наведено порівняння цих систем з метою дослідження компромісу між точністю моделі та її швидкістю в задачі виявлення об’єктів.

Швидкий YOLO — це найшвидший метод виявлення об'єктів на PASCAL; наскільки відомо, це найшвидший сповіщувач об'єктів. З 52.7% mAP він більш ніж удвічі точніший за попередні роботи з виявлення в реальному часі. YOLO висуває mAP до 63.4%, зберігаючи продуктивність у режимі реального часу.

Автори також тренують YOLO, використовуючи VGG-16. Ця модель є більш точною, але також значно повільнішою, ніж YOLO. Це корисно для порівняння з іншими системами виявлення, які покладаються на VGG-16, але оскільки він проходить повільніше, ніж у режимі реального часу, решта досліджень зосереджується на швидких моделях.

Найшвидший DPM ефективно прискорює DPM, не приносячи великих втрат mAP, але все одно не вистачає продуктивності в режимі реального часу

в 2 рази [40]. Він також обмежений порівняно низькою точністю DPM на виявлення порівняно з нейронно-мережевими підходами.

R-CNN мінус R замінює селективний пошук на пропозиції зі статичними обмежувальними рамками [62]. Хоча це набагато швидше, ніж R-CNN, йому все ще не вистачає швидкості реального часу і значно втрачає точність, оскільки у нього немає хороших пропозицій.

Faster R-CNN прискорює етап класифікації R-CNN, але він все ще покладається на вибірковий пошук, який може зайняти близько 2 секунд на зображення для створення пропозицій обмежувальних прямокутників. Таким чином, він має високий mAP, але при 0.5 кадрів в секунду це ще далеко від реального часу.

Нещодавній Faster R-CNN замінює селективний пошук нейронною мережею, щоб запропонувати обмежувальні коробки, подібні до Szegedy et al. [42]. У авторських тестах їх найточніша модель досягає 7 кадрів в секунду, а менша, менш точна — при 18 кадрів в секунду. Версія VGG-16 Faster R-CNN на 10 мАП вище, але також у 6 разів повільніше, ніж YOLO. Faster R-CNN Zeiler-Fergus — лише в 2,5 рази повільніше, ніж YOLO, але також менш точний.

Авторами підходу було порівняно властивості YOLO з Fast R-CNN на наборі даних VOC 2007, оскільки Fast R-CNN має один з найвищих показників детекторів на PASCAL і його виявлення є загальнодоступними. Автори використовували методологію та інструменти Hoiem et al. [49]. Для кожної категорії в тестовий час розглядаються N основних прогнозів для цієї категорії. Кожне передбачення або правильне, або його класифікують за типом помилки:

1. Правильно: правильний клас та IOU>0.5
2. Локалізація: правильний клас, 0.1<IOU<0.5
3. Подібні: клас схожий, IOU>0.1
4. Інше: клас неправильний, IOU>0.1

5. Фон: IOU<0.1 для будь-якого класу

На рис. 5.3 показано розподіл кожного типу помилок, усереднених для всіх 20 класів.

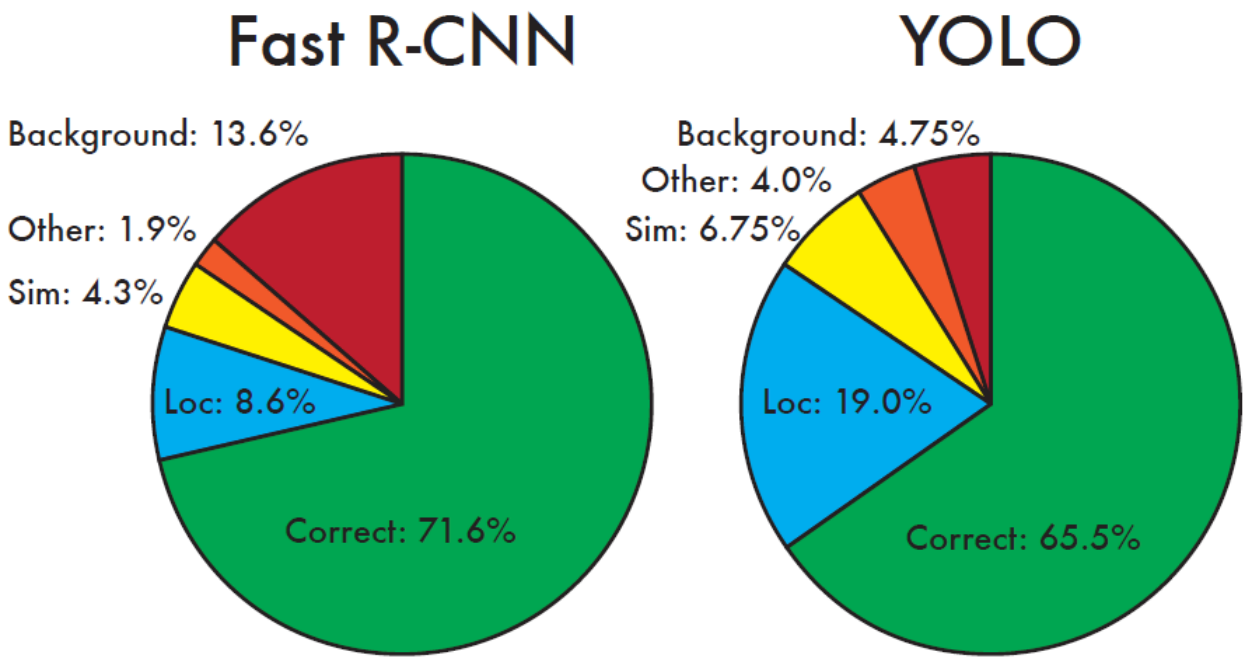


Рисунок 5.3 — Розподіл помилок

YOLO намагається правильно локалізувати об'єкти. Помилки локалізації припадають на більше помилок YOLO, ніж усі інші джерела разом. Fast R-CNN робить набагато менше помилок локалізації, але значно більше фонових помилок. 13,6% найкращих виявлень — це помилкові позитиви, які не містять жодних об'єктів. Fast R-CNN майже в 3 рази частіше прогнозує фонове виявлення, ніж YOLO.

В табл. 5.4 наведено чисельне порівняння результатів виявлення для YOLO та інших систем виявлення об’єктів.

Таблиця 5.4 — порівняння YOLO з іншими системами виявлення об’єктів

Real-Time Detectors	Train	mAP	FPS
100Hz DPM	2007	16.0	100
30Hz DPM	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
Less Than Real-Time			
Fastest DPM	2007	30.4	15
R-CNN Minus R	2007	53.5	6
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ZF	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

YOLO робить набагато менше фонових помилок, ніж Fast R-CNN. Використовуючи YOLO для усунення фонових виявлень від Fast R-CNN, отримують значне підвищення продуктивності. Для кожного обмежувального поля, яке прогнозує R-CNN, ми перевіряємо, чи YOLO прогнозує подібне поле. Якщо це так, ми надаємо цьому прогнозуванню підсилення, засноване на ймовірності, передбаченій YOLO, та перекритті між двома полями.

Найкраща модель Fast R-CNN досягає 71,8% mAP на тестовому наборі VOC 2007. У поєднанні з YOLO його mAP збільшується на 3,2% до 75,0%. Автори YOLO також спробували поєднати топ-модель Fast R-CNN з кількома іншими версіями Fast R-CNN. Ці ансамблі мали невеликий приріст mAP між 0.3 та 0.6% (табл. 5.5).

Таблиця 5.5 — Результати комбінування моделей на VOC 2007 наборі даних

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	66.9	72.4	0.6
Fast R-CNN (VGG-M)	59.2	72.4	0.6
Fast R-CNN (CaffeNet)	57.1	72.1	0.3
YOLO	63.4	75.0	3.2

На тестовому наборі VOC 2012 YOLO набирає 57,9% mAP. Це нижче, ніж сучасний стан технологій, ближче до оригінального R-CNN з використанням VGG-16. YOLO система бореться з дрібними предметами порівняно з найближчими конкурентами. У таких категоріях, як пляшки, вівці та телевізор/монітор YOLO на 8-10% нижче, ніж у R-CNN або Feature Edit. Однак на інших категоріях, таких як кіт і поїзд, YOLO досягає більш високих показників.

Комбінована модель Fast R-CNN + YOLO — один з найефективніших методів виявлення. Fast R-CNN отримує на 2,3% покращення від комбінації з YOLO. Посилення від YOLO — це не просто побічний продукт злияння моделей, оскільки від поєднання різних версій Fast R-CNN мало користі. Скоріше, саме тому, що YOLO робить різні види помилок у тестовий час, вона настільки ефективна для підвищення продуктивності Fast R-CNN.

На жаль, ця комбінація не має переваги від швидкості YOLO, оскільки кожна модель запускається окремо, а потім поєднуються результати. Однак, оскільки YOLO настільки швидкий, він не додає значного часу на обчислення порівняно з Fast R-CNN.

Автори також порівнювали YOLO з іншими системами виявлення наборів даних Picasso та People-Art. На рис. 5.4 та в табл. 5.6 показані порівняльні показники між YOLO та іншими методами виявлення. Для довідки, середня точність (Average Precision - AP) вимірюється на наборі даних

VOC 2007 по категорії «особа», де всі моделі навчаються лише за даними VOC 2007. На Picasso моделі навчаються на VOC 2012, тоді як на People-Art вони навчаються на VOC 2010.

R-CNN має високий рівень AP на VOC 2007. Однак R-CNN значно програє, коли застосовується до художнього твору. R-CNN використовує селективний пошук для пропозицій з обмежувальним вікном, налаштованим на природні зображення. Класифікаційний крок у R-CNN бачить лише невеликі регіони та потребує гарних пропозицій.

DPM (Deformable Part Model) добре підтримує свій AP при застосуванні до художнього твору. Попередня робота теоретизує, що DPM працює добре, оскільки має сильні просторові моделі форми та компонування об’єктів. Хоча DPM не погіршує стільки, скільки R-CNN, він показує нижчий AP.

YOLO має хороші показники на VOC 2007, і його AP деградує менше, ніж інші методи, коли застосовується до художнього твору. Як і DPM, YOLO моделює розміри та форму об'єктів, а також відносини між об'єктами та місцями, де об'єкти зазвичай з’являються. Художнє зображення та природні зображення дуже різняться на піксельному рівні, але вони схожі за розміром та формою предметів, тому YOLO все ще може передбачити хороші обмежувальні коробки та самі виявлення.

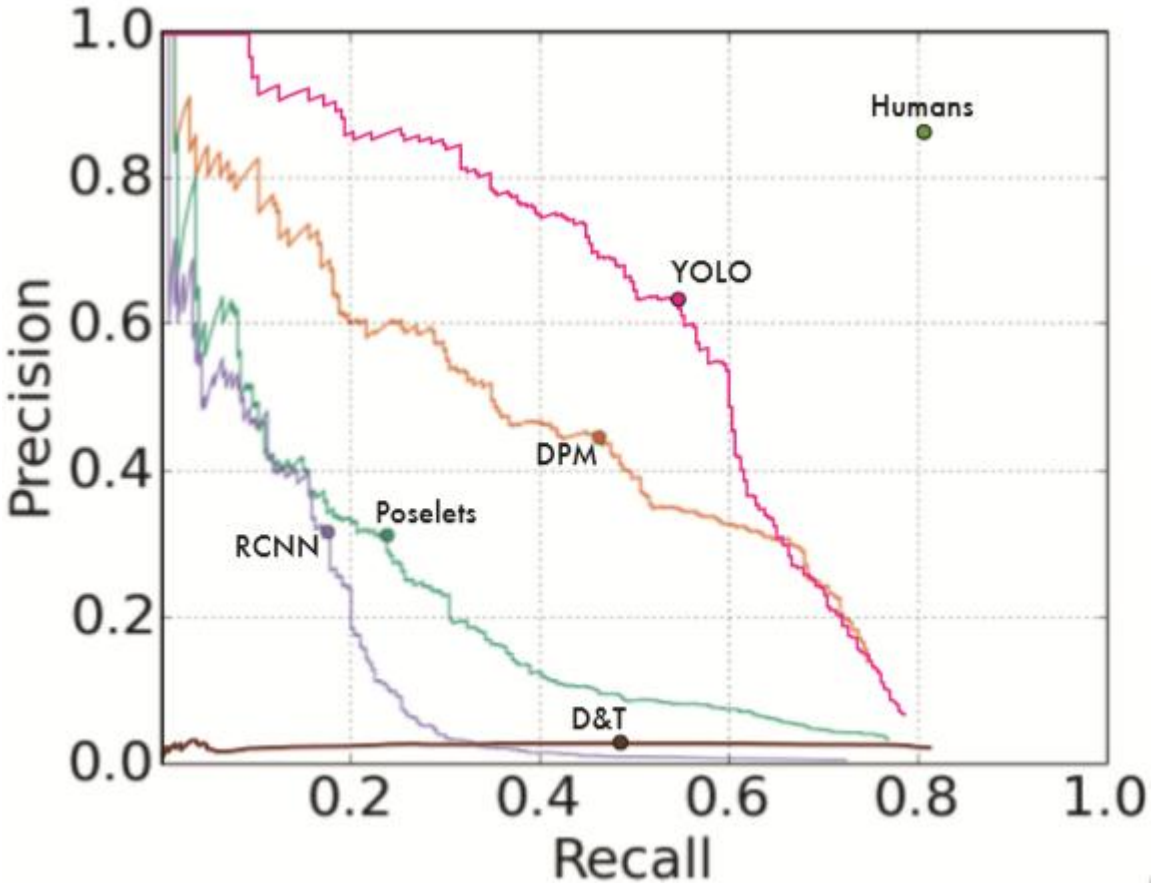


Рисунок 5.4 — Криві точності-відкликання на наборі даних Picasso

Таблиця 5.6 — Кількісні показники розглянутих моделей на різних наборах даних

	VOC 2007	Picasso		People-art
	AP	AP	Best F1	AP
YOLO	59.2	53.3	0.590	45
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets	36.5	17.8	0.271	
D&T	-	1.9	0.051	

5.4 Експеримент з комбінованою архітектурою

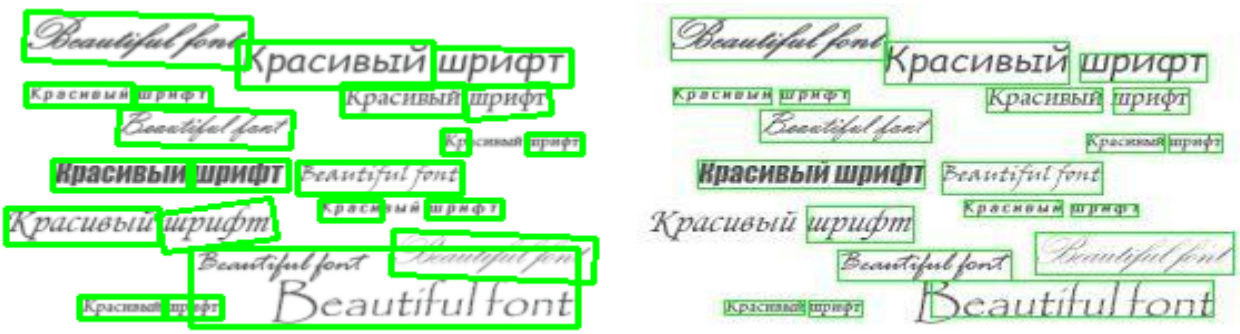
Тренування комбінованого підходу відбувалося на на наборі SynthText протягом 10 днів на відеокарті NVIDIA TITAN X. В якості оптимізатора був використаний ADAM, початкове значення коефіцієнта швидкості навчання було встановлене /=0.05.

Порівняльні результати можна побачити на рис. 5.5 — 5.6. Як видно на рис 5.5, комбінований підхід дає кращі результати, особливо на тексті знизу, який був наданий в порівняно гіршому дозволі, ніж основний текст. Це говорить про те, що лише теплових карт недостатньо для ефективного виявлення тексту.

а) б)

Рисунок 5.5 — Порівняння результатів виявлення тексту з однаковим шрифтом з використанням мереж: а) CRAFT; б) комбінованого підходу

На рис. 5.6 можна побачити, що обидва підходи впоралися з задачею не ідеально. Але можна побачити, що CRAFT схиляється групувати слова з різних строк в одне обмежувальне поле, що обумовлене алгоритмом післяобробки.



а) б)

Рисунок 5.6 — Порівняння результатів виявлення тексту з різними шрифтами з використанням мереж: а) CRAFT; б) комбінованого підходу

Експеримент показав, що комбінований підхід є багатообіцяючою конструкцією. Новий підхід прогнозує точніші обмежувальні поля та робить менше помилок, пов’язаних з групуванням різних слів у одне обмежувальне поле. Слід відмітити, що навчання комбінованого підходу пройшло не повністю, тобто нейронна мережа показує не найкращі з потенційних результатів. Це пов’язано з тим, що модуль від CRAFT є набагато більшим, ніж частина від YOLO, тож остання починає перенавчатися до даних, в той час коли модуль від CRAFT ще тільки починає збігатись.

Це наводить на думку, що треба або полегшувати модуль CRAFT з точки зору кількості навчальних параметрів, або ж заморожувати модуль від YOLO під час останньої стадії навчання.

ВИСНОВКИ

В ході досліджень було розглянуто існуючі методи та підходи в задачі виявлення тексту в природних зображеннях. Переважно задача виявлення об’єктів на зображенні в цілому, а тексту — зокрема, вирішується двома підходами: однопрохідним та багатопрохідним. В однопрохідному підході нейронна мережа генерує результати виявлення на основі мап ознак, отриманих з основної архітектурної конструкції. Прикладами такого підходу є архітектури YOLO та SSD. В багатопрохідному підході на відміну від однопрохідного результати виявлення отримуються з мап ознак, отриманих спеціальним модулем, як то реалізовано в RCNN підходах.

З іншого боку, існує ще один підхід, в якому результати виявлення представлені тепловою картою об’єктів, а не координатами обмежувальних полей. В атестаційній роботі були проведені дослідження одного з таких підходів, який називається CRAFT.

Для подолання недоліків розглянутих підходів був розроблений та запропонований новий підхід, який має переваги однопрохідних методів в простоті та ефективності архітектури з одного боку, а з іншого — переваги від підходу CRAFT, який зосереджується на виявленні теплових карт символів та зв’язків між ними.

Було проведене експериментальне дослідження запропонованого підходу, яке показало його ефективність. Майбутні дослідження можуть бути спрямовані в сторону оптимізації архітектури запропонованого підходу, оскільки в модулі CRAFT використовується досить стара основа — VGG-16, яка є неефективною з точки зору кількості навчальних параметрів, активацій та точності класифікації на наборі даних ImageNet. Ці фактори знижують швидкість та ефективність навчання та результати виявлення.

ПЕРЕЛІК ПОСИЛАНЬ

1 D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. In AAAI, 2018. 1, 2, 6

2 P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In ICCV, volume 6, 2017. 1, 2, 6

3 T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun. An end-to-end textspotter with explicit alignment and attention. In CVPR, pages 5020–5029, 2018. 1, 2, 6, 7

4 H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding. Wordsup: Exploiting word annotations for character based text detection. In ICCV, 2017. 1, 2, 5, 6

5 Zh. Hu, Ye. Bodyanskiy, N. Kulishova, O. Tyshchenko A Multidimensional Extended Neo-Fuzzy Neuron for Facial Expression Recognition / Int. Journal of Intelligent Systems and Applications (USA), vol. 9, No. 9, pp. 29-36, 2017.

6 X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. In CVPR, pages 5676–5685, 2018. 1, 2, 6, 7

7 S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. arXiv preprint arXiv:1807.01544, 2018. 1, 2, 6

8 P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. arXiv preprint arXiv:1807.02242, 2018. 1, 2, 6, 7

9 B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. In CVPR, pages 3482– 3490. IEEE, 2017. 1, 2, 5, 6

10 Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9365–9374, 2019.

11 D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In ICDAR, pages 1156–1160. IEEE, 2015. 1

12 C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In CVPR, pages 1083–1090. IEEE, 2012. 2

13 L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng. Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170, 2017. 2, 6, 11

14 C. K. Ch'ng and C. S. Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In ICDAR, volume 1, pages 935–942. IEEE, 2017. 2

15 J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. Image and Vision Computing, 22(10):761–767, 2004. 2

16 B. Epshtein, E.Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In CVPR, pages 2963–2970. IEEE, 2010. 2

17 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In ECCV, pages 21–37. Springer, 2016. 2

18 S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. PAMI, (6):1137–1149, 2017. 2

19 J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, pages 3431– 3440, 2015. 2

20 M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In AAAI, pages 4161–4167, 2017. 2

21 Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In CVPR, pages 3454– 3461, 2017. 2

22 M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai. Rotation sensitive regression for oriented scene text detection. In CVPR, pages 5909–5918, 2018. 2, 6

23 Ye. Bodyanskiy, Yu. Zaychenko, N. Kulishova, G. Hamidov. Spline-Orthogonal Extended Neo-Fuzzy Neuron / Proc. of 2019 12th International

Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2019) , 19 - 21 October 2019, Huaqiao, Suzhou, China – 2019, DOI: 10.1109/CISP-BMEI48845.2019.8965840

24 N. Kulishova , Ye. Bodyanskiy, I. Pliss. The Extended Generalized Neo-Fuzzy Network and its Online Learning in Image Recognition Problem / Proc. of 2019 10th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Sept. 18-21, 2019. Metz, France. Pp. 34-40. DOI: 10.1109/IDAACS.2019.8924367

25 Ye. Bodyanskiy, N. Kulishova, O. Chala, "The Extended Multidimensional Neo-Fuzzy System and its Fast Learning in Pattern Recognition Tasks," Data, №3, 2018, pp. 63 – 73. doi:10.3390/data3040063

26 D. He, X. Yang ,C. Liang, Z. Zhou, G. Alexander, I. Ororbia, D. Kifer, and C. L. Giles. Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In CVPR, pages 474–483, 2017. 2

27 C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002, 2016. 2, 6

28 Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In CVPR, pages 4159–4167, 2016. 2, 6

29 C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In Computer vision, 1998. sixth international conference on, pages 555–562. IEEE, 1998. 4

30 D. G. Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee, 1999. 4

31 N. Dalal and B.Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005. 4, 8

32 P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010. 1, 4

33 M. Lin, Q. Chen, and S. Yan. Network in network. CoRR, abs/1312.4400, 2013. 2

34 P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229, 2013. 4, 5

35 J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013. 4

36 R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015. 2, 5, 6, 7

37 S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 5, 6, 7

38 T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik, et al. Fast, accurate detection of 100,000 object classes on a single machine. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1814–1821. IEEE, 2013. 5

39 M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In Computer Vision–ECCV 2014, pages 65–79. Springer, 2014. 5, 6

40 J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2497–2504. IEEE, 2014. 5, 6

41 P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004. 5

42 D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014. 5, 6

43 Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229.

44 J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. CoRR, abs/1412.3128, 2014. 5

45 K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 3

46 O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, pages 234–241. Springer, 2015. 3

47 A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, pages 483–499. Springer, 2016. 3

48 L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. PAMI, (6):583–598, 1991. 4

49 D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In Computer Vision–ECCV 2012, pages 340–353. Springer, 2012. 6

50 C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014. 2

51 O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015. 3

52 D. Mishkin. Models accuracy on imagenet 2012 val. URL: https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val. Accessed: 2015-10-2. 3

53 J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016. 3

54 S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, abs/1504.06066, 2015. 3, 7

55 G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012. 4

56 Explaining and Harnessing Adversarial Examples, Goodfellow et al, ICLR 2015.

57 Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in The AAAI Conference on Artificial Intelligence (AAAI), 2020.

58 J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

59 H. Cai, Q. Wu, T. Corradi, and P. Hall. The cross-depiction problem: Computer vision algorithms for recognising objects in artwork and in photographs. arXiv preprint arXiv:1505.00110, 2015. 7

60 S. Ginosar, D. Haas, T. Brown, and J. Malik. Detecting people in cubist art. In Computer Vision-ECCV 2014 Workshops, pages 101–116. Springer, 2014. 7

61 H. Cai, Q. Wu, T. Corradi, and P. Hall.The cross-depiction problem: Computer vision algorithms for recognizing objects in artwork and in photographs. arXiv preprint arXiv:1505.00110, 2015. 7

62 K. Lenc and A. Vedaldi. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015. 5, 6