Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

(повна назва)

Кафедра Програмної інженерії

(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти – другий (магістерський)

Дослідження алгоритмів знаходження обличчя на цифровому зображенні

(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-3

Богомолов О.Є.

(прізвище, ініціали)

спеціальності 121 – Інженерія програмного забезпечення

(код і повна назва спеціальності)

Освітньо-наукової програми

(тип програми)

Інженерія програмного забезпечення

(повна назва освітньої програми)

Керівник к.т.н., доц. Назаров О.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

20__ р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри_____

(підпис)

«____»_____20___р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Богомолову Олександру Євгенійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження алгоритмів знаходження обличчя на цифровому зображенні

затверджена наказом університету від «27» березня 2020р. № 473 Ст

заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

19 травня 2019 р.

3. Вихідні дані до роботи згорткові нейронні мережі, набір даних WIDER FACE, TensorFlow, модель YOLO, модель MTCNN, точність, повнота, середня точність.

4. Перелік питань, що потрібно опрацювати в роботі вступ, аналіз предметної галузі, опис існуючих алгоритмів, модель R-CNN, модель YOLO, модель MTCNN, формулювання гіпотез, опис і аналіз результатів експерименту, висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка *
1	Аналіз предметної галузі	27 березня 2020 р.	
2	Огляд існуючих методів	3 квітня 2020 р.	
3	Підготовка до експерименту	10 квітня 2020 р.	
4	Проведення експерименту	17 квітня 2020 р.	
5	Аналіз результатів експерименту	24 квітня 2020 р.	
6	Підготовка пояснювальної записки	8 травня 2020 р.	
7	Попередній захист	11 травня 2020 р.	
8	Нормоконтроль, рецензування	12 травня 2020 р.	
9	Занесення диплома в електронний архів	16 травня 2020 р.	
10	Допуск до захисту у зав. кафедри	18 травня 2020 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ _____ 2019 р.

Студент _____Богомолов О.Є

(підпис)

Керівник роботи _____ к.т.н, доц. Назаров О.С.

(підпис)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 58 с., 16 рисунків, 4 таблиці, 25 джерел.

АЛГОРИТМИ ЗНАХОДЖЕННЯ ОБЛИЧЧЯ, ЗГОРТКОВІЙ НЕЙРОННІ МЕРЕЖІ, СЕРЕДНЯ ТОЧНІСТЬ, MTCNN, R-CNN, TENSORFLOW, YOLO.

Об’єктом дослідження є нейромережеві алгоритми знаходження обличчя на цифрових зображеннях.

Метою роботи є визначення методу знаходження обличчя на цифрових фотографіях в оптимальних і неоптимальних умовах.

В результаті виконання дослідження були сформульовані рекомендації щодо застосування досліджуваних алгоритмів для задач з різного роду особливостями, такими як обмеженням часу роботи і особливостями вхідних зображень.

Також було запропоновано метод, заснований на комбінації цих алгоритмів, який ефективніше знаходить обличчя, які є частково перекритими, або частково перебувають за межами кадру.

AVERAGE PRECISION, CONVOLUTIONAL NEURAL NETWORKS, FACE DETECTION ALGORITHMS, MTCNN, R-CNN, TENSORFLOW, YOLO.

The object of the study are the neural network-based face detection algorithms.

The purpose of the work is to determine the optimal method of detecting faces in digital photographs under optimal and suboptimal conditions.

As the result of the study, the recommendations for usage of studied algorithms for solving tasks with different features, such as time constraints or input images features, were determined.

A method based on a combination of studied algorithms has also been proposed. This method finds faces that are partially overlapped by other objects or partially outside the image more effectively.

ЗМІСТ

ВСТУП

Завдання знаходження облич на цифрових зображеннях є актуальною досить довгий час. Також важливим є завдання знаходження на зображення облич, частково перекритих іншими об'єктами, або облич, які частково перебувають за межами кадру. Методи знаходження осіб мають широке застосування в різних сферах, таких як безпека, охорона здоров’я і маркетинг. В останні роки, завдяки стрімкому розвитку глибоких нейронних мереж, з’явилися нові методи вирішення цих задач. Різні моделі мають різні переваги і недоліки, і через їхню різноманітність, обрати модель для вирішення специфічного завдання досить непросто.

Ця робота продовжує дослідження кафедри Програмної інженерії в області комп'ютерного зору. [1-5]

Метою роботи є визначення рекомендацій щодо застосування досліджуваних моделей для вирішення різних завдань, а саме знаходження облич на зображеннях в реальному часі, знаходження на зображені облич, що не є перекритими іншими об'єктами і які є повністю видимими, і знаходження облич разом з особливими точками. Також важливою задачею роботи є визначення оптимального методу для знаходження облич, які перекриті іншими об’єктами, або облич, які частково знаходяться за межами кадру.

Ще одним завданням роботи є порівняння ефективності загальних методів для знаходження будь-яких об’єктів в задачах знаходження облич з методами, спеціально розробленими для знаходження облич на зображеннях.

Об’єктом дослідження є непромережені технології і методи знаходження об’єктів на цифрових зображеннях.

Предметом дослідження є непромережені алгоритми для знаходження осіб на цифрових зображеннях.

Методом дослідження є проведення експерименту, використовуючи набір даних з різними зображеннями. Набір даних включає зображення з обличчями під

різними кутами і в різних умовах освітленості, обличчями великого і маленького розміру відносно вхідного зображення, а також з обличчями, повністю видимими, і обличчями, частково перекритими іншими об'єктами, або з обличчями, які частково знаходяться за межами кадру. Для порівняння різних алгоритмів, необхідно визначити чисельну метрику, яка оцінює ефективність кожного з методів. У цій роботі в якості метрики використовується метрика середньої точності. Вона дозволяє об'єктивно порівнювати різні алгоритми для знаходження об'єктів на цифрових зображеннях.

Елементом наукової новизни в роботі є пропозиція нового методу знаходження частково закритих іншими об'єктами облич. Цей метод заснований на комбінації двох досліджуваних нейромережевих моделей. Перша модель обмежує область, в якій знаходиться людина, а друга модель визначає обличчя в цій обмеженій області. Обмеження області пошуку дозволяє підвищити ефективність знаходження облич невеликого розміру і облич, перекритих іншими об'єктами.

В результаті роботи були також сформульовані рекомендації щодо використання досліджуваних моделей для різних завдань.

Результати проведеного експерименту також були подані до публікації в матеріалах конференції Data Stream Mining & Processing (DSMP) 2020.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Технології знаходження облич на цифрових фотографіях існують досить довго і мають багато застосувань в різних сферах і галузях. Сьогодні основними галузями, які використовують технології розпізнавання облич, є безпека, охорона здоров'я і маркетинг. [10, 11]

В галузі безпеки технології знаходження облич використовуються досить довго. Перш за все, технології розпізнавання осіб використовуються в системах спостереження, які використовуються як і державними організаціями, так і приватними фірмами. Основним застосуванням технології в таких системах є аналіз потоків людей в багатолюдних місцях, таких як аеропорти, центральні вулиці міста, митниця і інші. Система аналізує потік людей, і по знайденим обличчям ідентифікує людей, які проходять через ділянку, за якою здійснюється спостереження. Отримані дані можна використовувати як і для благих цілей, таких як пошуку зниклих людей або злочинців, так і для масової стеження за населенням держави.

У приватних організаціях технологія знаходження облич для систем безпеки використовується, як і в державних організаціях, в основному для ідентифікації людей. Цікавим застосуванням технологій розпізнавання осіб в системах безпеки є виявлення шахрайства. За допомогою камер безпеки, встановлених в закладі, розпізнаються обличчя відвідувачів. Потім відбувається пошук по базі даних з людьми, які в минулому здійснювали правопорушення або яких заборонено обслуговувати в конкретному закладі. Також відбувається пошук по відеоархіву. У разі виявлення відвідувача як потенційного правопорушника, система робить звернення в службу безпеки закладу. Також технології розпізнавання облич застосовують в різних системах контролю доступу.

В останні роки технології розпізнавання облич також широко використовується в мобільному сегменті, де вони використовується для прийняття рішення про розблокування пристрою, якщо їм в даний момент користується його

власник. Це застосування технологій розпізнавання облич особливо примітним, тому що для ідентифікації власника пристрою використовується в більшості випадків не звичайна цифрова фотографія, а тривимірна модель обличчя людини, який в даний момент користується пристроєм, яка була отримана за допомогою спеціальної інфрачервоної камери. Це дозволяє підвищити точність системи розпізнавання особи власника пристрою, а також реагувати на зміну зовнішності власника пристрою.

Технології розпізнавання облич також застосовуються в сфері охорони здоров'я. Одним з найпопулярніших застосувань цих технологій в сфері охорони здоров'я є використання технологій розпізнавання облич для забезпечення безпеки лікарень. Система безпеки сканує обличчя всіх людей, що входять до лікарні, і порівнює їх з обличчями людей, помічених як небезпечних для цієї лікарні. Така технологія може бути використана для знаходження наркозалежних людей, які приходять в лікарню з метою отримання наркотичних речовин, або для знаходження людей, які були помічені як небезпечні для лікарні. Також технології розпізнавання облич можуть застосовуватися для виявлення людей, який представляються пацієнтами, але насправді не потребують медичної допомоги, з метою отримання лікарських препаратів.

Важливим застосуванням технологій виявлення осіб в охороні здоров'я є розпізнавання емоцій пацієнта протягом його перебування в лікарні. Це допомагає визначити, який пацієнт потребує більшої уваги зі сторони персоналу закладу охорони здоров’я, тому що він відчуває біль або сум.

Технології розпізнавання обличчя в маркетингу допомагають у просуванні товару або послуги на ринку. Наприклад, розпізнавання статі і віку людини по його обличчю допомагає бізнесу краще підібрати продукт для цього клієнта.

Надаючи можливість підрахувати конкретне число клієнтів, які відвідують магазини підприємства, технологія розпізнавання облич може допомогти з'ясувати причину низьких продажів, або, навпаки, причину високих доходів.

Більш того, технології розпізнавання обличчя можуть надати велику допомогу власникам супермаркетів або мереж ресторанів навіть до того, як вони

відкриють магазин або ресторан в новій місцевості. Якщо вони впроваджені досить довго до того, коли підприємство відкриється, вони можуть підказати, наскільки підходить публіка, яка відвідує цю місцевість, для досягнення фінансових цілей підприємства. Як мінімум, можна з'ясувати, чи знаходиться достатня кількість людей в цій місцевості у відповідний час, щоб досягти певного рівня продажів.

2 ОПИС ІСНУЮЧИХ АЛГОРИТМІВ

Проблема розпізнавання обличчя стала дуже важливою у галузі біометричної ідентифікації. Виявлення обличчя в цифрових зображеннях має багато застосувань для вирішення реальних проблем, включаючи розваги, охорону здоров'я, безпеку тощо. На сьогоднішній день існує маса підходів та моделей для вирішення цих завдань. Хоча старіші алгоритми, такі як алгоритм Віола-Джонса, виконують завдання розпізнавання обличчя швидко і точно в близьких до оптимальних умовах, їх ефективність знижується, коли вхідні зображення спотворені, занадто темні або занадто світлі або коли обличчя з’являються на зображенні під різними кутами. Розвиток глибокого навчання та глибоких згорткових нейронних мереж дозволив дослідникам створити складні алгоритми машинного навчання на основі нейронних мереж, які при належній підготовці набагато краще і надійніше працюють на зображеннях з обличчями в різних умовах, включаючи умови, коли обличчя знаходиться під різними кутами та в занадто темних або занадто світлих частинах фотографії. Крім того, кожен алгоритм на основі нейронної мережі має свої особливості та характеристики, які визначають його ефективність при застосуванні для різних типів вхідних фотографій.

Проблема розпізнавання обличчя – це підмножина більш загальної проблеми, проблеми розпізнавання об'єктів. Існують дві родини алгоритмів на основі нейронних мереж, які найчастіше використовуються для знаходження об'єктів на фотографіях або кадрів з відео: R-CNN алгоритми, які використовують нейронні мережі для пропозиції регіонів (RPN) для ідентифікації ділянок на зображенні, де може бути об'єкт, та YOLO (абревіатура для You Only Look Once, ті лише дивишся один раз), що є більш швидким, але менш точним алгоритмом, ніж R-CNN. Існують й інші підходи та алгоритми знаходження об'єктів, однак два наведених вище підходи є найбільш розповсюдженими в галузі. Кожен з цих підходів також інкрементно вдосконалюється для підвищення точності та усунення недоліків.

Також існує нейромережева модель, яка була розроблена спеціально для вирішення проблеми розпізнавання обличчя, MTCNN. Модель MTCNN складається з трьох окремих нейронних мереж - P-Net, R-Net та O-Net, які визначають не тільки обмежувальний прямокутник навколо обличчя, але й координат розташування очей, носа та рота.

2.1 Модель R-CNN

Різниця між алгоритмами для знаходження об’єктів на зображенні і алгоритмами для класифікації зображень є те, що в разі знаходження об'єкта необхідно отримати координати прямокутника, що містить шуканий об'єкт. Більш того, часто на одному зображенні необхідно знайти кілька прямокутників, в разі якщо на зображенні присутні кілька шуканих об’єктів, число яких заздалегідь невідомо. Причиною неможливості використання звичайних згорткових мереж для знаходження об’єктів на зображенні є те, що розмірність вихідного шару не є фіксованою, тому що на зображенні можуть бути довільне число об’єктів, які потрібно знайти. Найпростішим рішенням цієї проблеми є визначення декількох різних регіонів зображення, і використання звичайної згорткової нейронної мережі для класифікації присутності того або іншого об'єкта в цьому регіоні. Проблема в цьому підході полягає в тому, що об'єкти на зображенні можуть знаходиться в різних місцях і можуть мати різні пропорції. Через це виникає необхідність аналізувати величезну кількість регіонів, і обчислювальна складність алгоритму дуже сильно зростає. Такі алгоритми як R-CNN та YOLO були розроблені саме для того щоб швидше знаходити подібні регіони.

Для вирішення цієї проблеми була розроблена модель R-CNN. [8] Було запропоновано використання алгоритму виборчого пошуку (selective search), щоб виділити рівно 2000 регіонів з зображення і назвати їх пропозиціями регіонів. [9]

Тепер, замість того щоб класифікувати величезну кількість регіонів, потрібно обробити всього 2000.

Алгоритм виборчого пошуку працює наступним чином:

– генерується початкова суб-сегментація, генерується велика кількість регіонів-кандидатів;

– використовується жадібний алгоритм для рекурсивного комбінування схожих регіонів в великі однакові регіони;

– згенеровані регіони використовуються для отримання остаточних пропозицій регіонів (див. рис. 2.1).

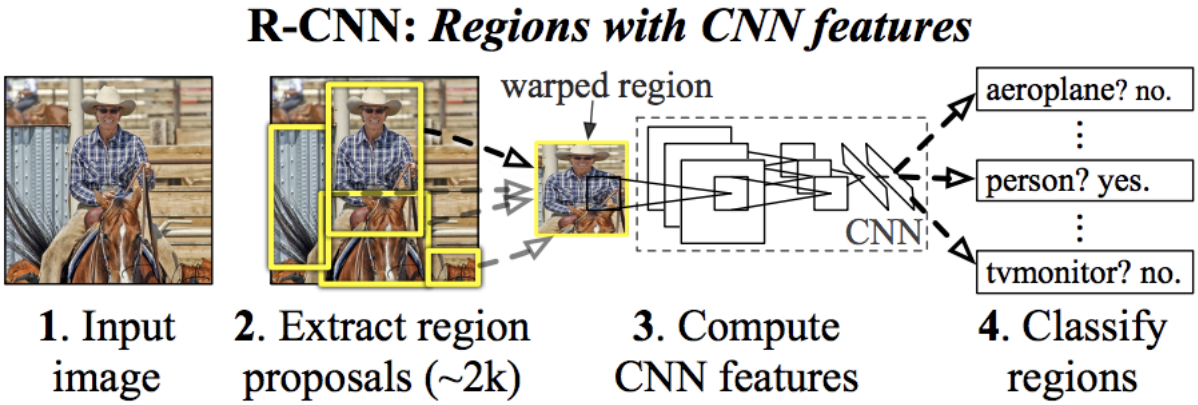

Рисунок 2.1 – Принцип роботи моделі R-CNN [8]

Ці 2000 запропонованих регіонів перетворюються в квадрати і передаються в згорткову нейронну мережу, яка виводить 4096-розмірний вектор. Цей вектор в свою чергу передається в SVM, інший алгоритм машинного навчання, який класифікує присутність об'єкта в запропонованому регіоні. Алгоритм також пророкує 4 значення координат, які представляють кордони виявленого об’єкту.

У першого варіанту моделі R-CNN є такі недоліки:

– як і раніше навчання нейронної мережі займає дуже багато часу, тому що для кожного зображення потрібно класифікувати 2000 запропонованих регіонів;

– неможливо використовувати цю модель в реальному часі, тому що обробка одного зображення займає приблизно 47 секунд;

– алгоритм виборчого пошуку не є алгоритмом машинного навчання, а значить навчання моделі на етапі генерації пропозицій регіонів не відбувається. Це може привести до генерації поганих пропозицій регіонів.

У наступному алгоритмі, Faster R-CNN, були вирішені деякі недоліки попереднього підходу. [10] Замість передачі в згорткову нейронну мережу пропозицій регіонів, вхідне зображення передається в згорткову нейронну мережу, яка генерує карту ознак. Використовуючи карту ознак, пропозиції регіонів ідентифікуються, перетворюються в квадрати, і за допомогою RoI пулінгу трансформуються в вектори фіксованого розміру, щоб передати їх в наступні повністю зв'язані шари, які використовуються для класифікації об’єкта, який знаходиться в запропонованому регіоні, і розрахунку координат обмежувальної рамки (див. рис. 2.2).

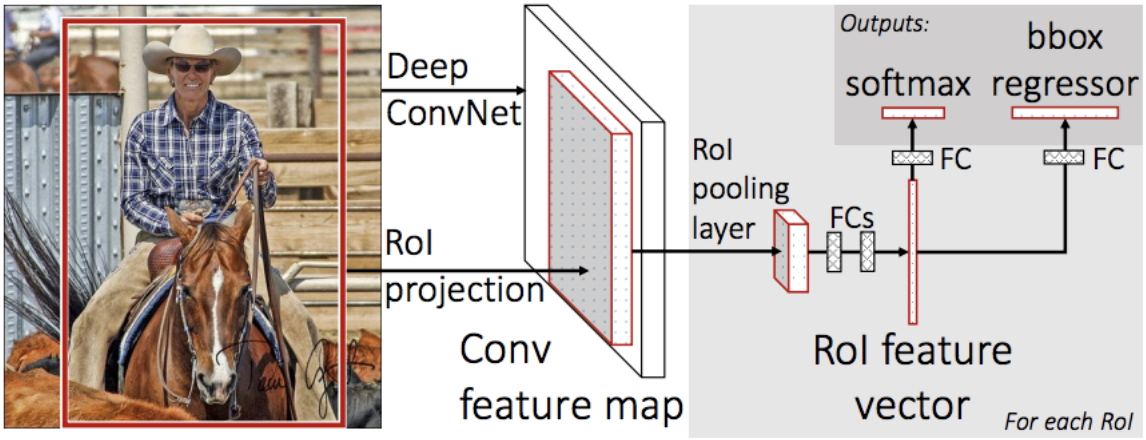

Рисунок 2.2 – Принцип роботи моделі Fast R-CNN [10]

Fast R-CNN швидше R-CNN тому, що немає необхідності кожен раз передавати 2000 пропозицій регіонів в згорткову нейронну мережу. Замість цього, операція згортки відбувається тільки один раз за зображення, і за допомогою неї генерується карта ознак.

Обидва наведених вище алгоритму використовують виборчий пошук для знаходження пропозицій регіонів. Виборчий пошук – повільний процес, який знижує загальну продуктивність нейронної мережі. Наступна версія, Faster R-CNN,

містить алгоритм для знаходження об'єктів, який прибирає необхідність в алгоритмі виборчого пошуку і дозволяє нейронної мережі вивчати знаходження пропозицій регіонів самостійно. [11]

Подібно Fast R-CNN, зображення передається в якості введення в згорткову нейронну мережу, яка виводить карту ознак. Замість використання алгоритму виборчого пошуку, окрема нейронна мережа використовується для знаходження пропозицій регіонів. [12] Знайдені пропозиції регіонів трансформуються за допомогою шару RoI пулінгу в вектор фіксованої довжини, який потім використовується для класифікації зображення, що знаходиться всередині запропонованого регіону, і для визначення обмежувальних рамок (див. рис. 2.3).

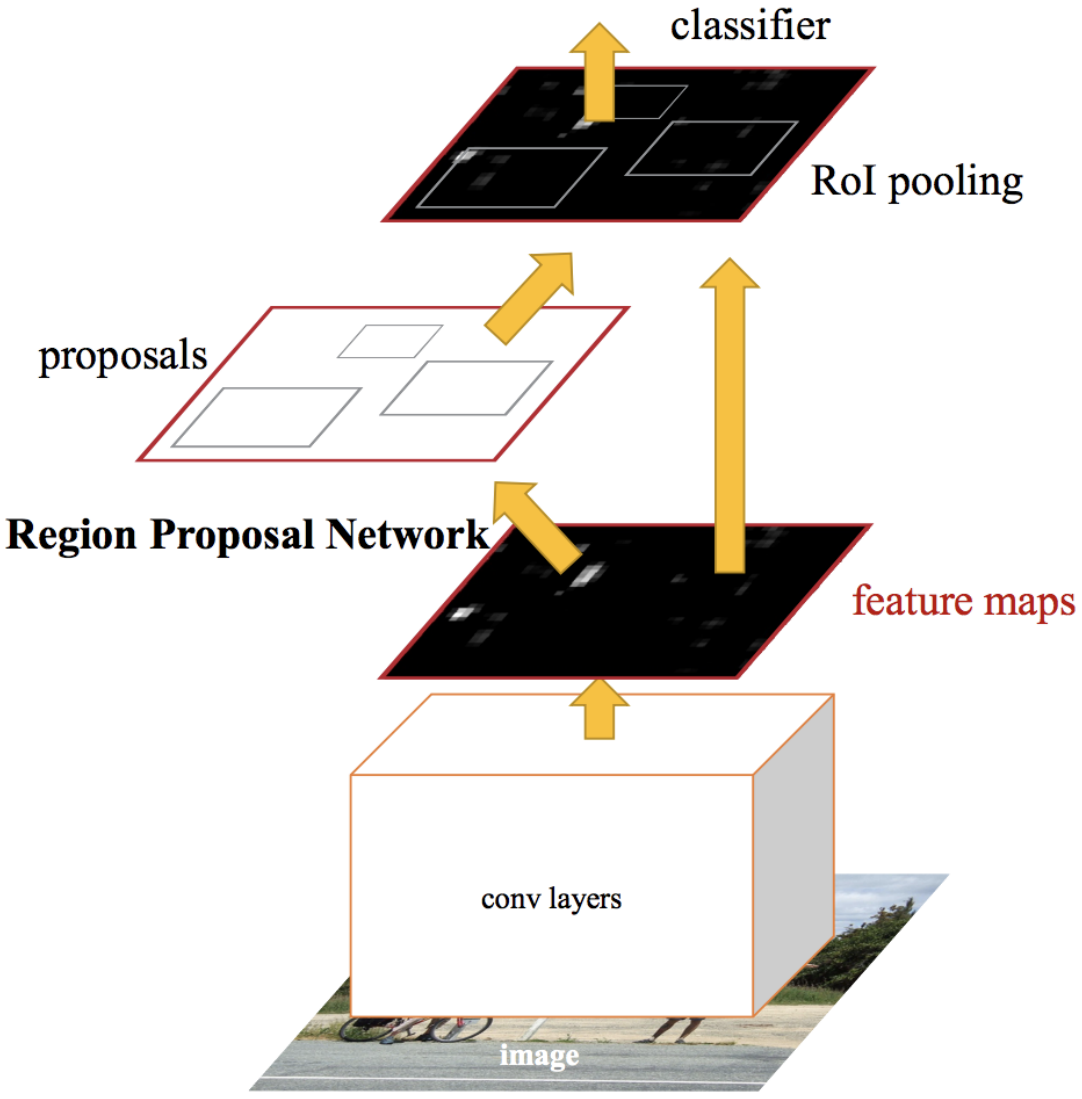

Рисунок 2.3 – Принцип роботи моделі Faster R-CNN [11]

Faster R-CNN працює значно швидше за своїх попередників, R-CNN і Faster R-CNN. Через це цю модель можна використовувати для знаходження об'єктів в реальному часі.

На жаль, модель Faster R-CNN все ще вимагає великих обчислювальних ресурсів для ефективного навчання, тому через відсутність обчислювальних ресурсів Faster R-CNN не бере участі в експерименті, поставленого в рамках цієї роботи.

2.2 Модель YOLO

YOLO є іншою групою нейромережевих моделей для знаходження об'єктів на зображенні. Як і моделі R-CNN, модель YOLO теж розвивалася послідовно. Модель YOLO фокусується на швидкому розпізнаванні об’єктів на зображенні, жертвуючи точністю роботи на догоду швидкості. [13] Це дозволяє моделі YOLO обробляти зображення значно швидше моделі R-CNN, що робить її хорошим вибором для випадків, коли потрібна швидка обробка зображень, наприклад, для знаходження об’єктів на потоковому відео.

Модель YOLO має три версії, YOLOv1, YOLOv2 і YOLOv3.

Модель YOLO працює за наступним принципом. YOLO розділяє вхідне зображення в сітку $S \times S$. Кожен осередок сітки відповідає за знаходження виключно одного об’єкта. Кожен осередок визначає фіксовану кількість обмежувальних рамок.

Таким чином, кожна клітинка:

– визначає B обмежувальних рамок і кожна рамка має одну оцінку впевненості;

– знаходить один об'єкт незалежно від кількості обмежувальних рамок B;

– прогнозує C умовних класових ймовірностей (одну на кожен клас об'єкта, які повинна знайти модель).

Кожна рамка, яку прогнозує модель, містить 5 елементів: (x, y, w, h) і оцінку впевненості. Оцінка впевненості відображає наскільки ймовірно рамка містить об’єкт і наскільки точними є координати обмежувальної рамки. Ширина w і висота h обмежувальної рамки нормалізуються щодо ширини і висоти зображення. x і y представляють собою зміщення відносно відповідного осередку. Отже, x, y, w і h приймають значення від 0 до 1. Умовна класова ймовірність – це ймовірність того, що знайдений об’єкт належить до конкретного класу (одна ймовірність для кожного класу для кожного осередку).

Наприклад, якщо модель ділить зображення на 7 осередків по горизонталі і вертикалі, кожна клітинка прогнозує 2 обмежувальних рамки, і модель знаходить 20 класів об’єктів, то розмірність виводу моделі становитиме $(S, S, B \times 5 + C) = (7,7,2 \times 5 + 20) = (7,7,30)$.

Основною концепцією YOLO є побудова згорткової нейронної мережі, яка виводить тензор розмірністю (7,7,30). Модель використовує згорткову нейронну мережу щоб знизити просторову розмірність до 7×7 з 1024 вихідними каналами в кожному положенні. YOLO виконує лінійну регресію використовуючи два повністю з'єднаних шару щоб зробити прогноз обмежувальних рамок розмірністю (7,7,2). Щоб зробити остаточне прогнозування, модель вибирає обмежувальні рамки з великою оцінкою впевненості (див. рис. 2.4).

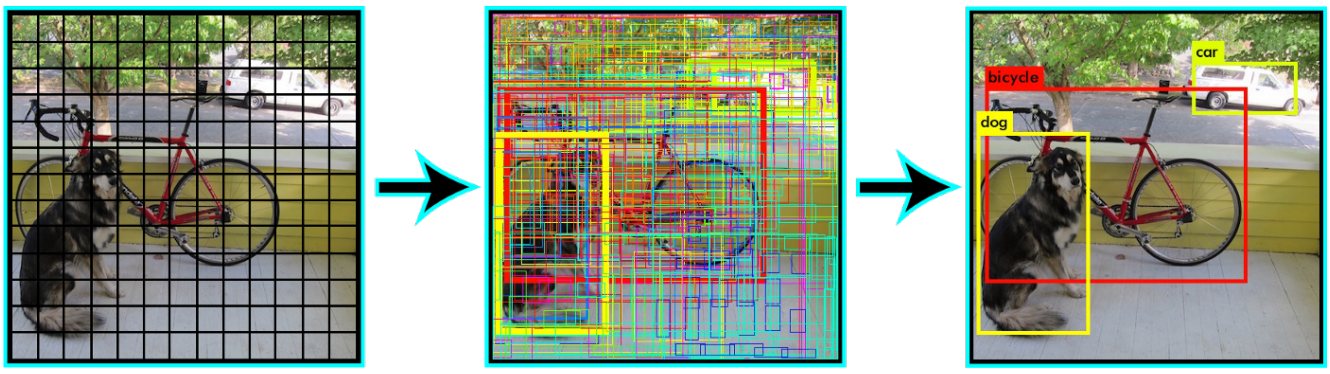

Рисунок 2.4 – Принцип роботи YOLO [13]

Оцінка впевненості класу для кожної знайденої обмежувальної рамки розраховується як добуток оцінки впевненості обмежувальної рамки і умовної ймовірності класу.

Архітектура нейронної мережі YOLO містить 24 згортальних шарів, за якими слідують два повністю з’єднаних шару. Останній згортальний шар виводить тензор розмірністю (7,7,1024). Цей тензор потім вирівнюється. Використовуючи два повністю з'єднаних шару як різновид лінійної регресії, модель виводить $(7 \times 7 \times 30)$ параметрів і потім трансформує їх у (7,7,30), тобто по 2 обмежувальних рамки на ділянку зображення (див. рис. 2.5).

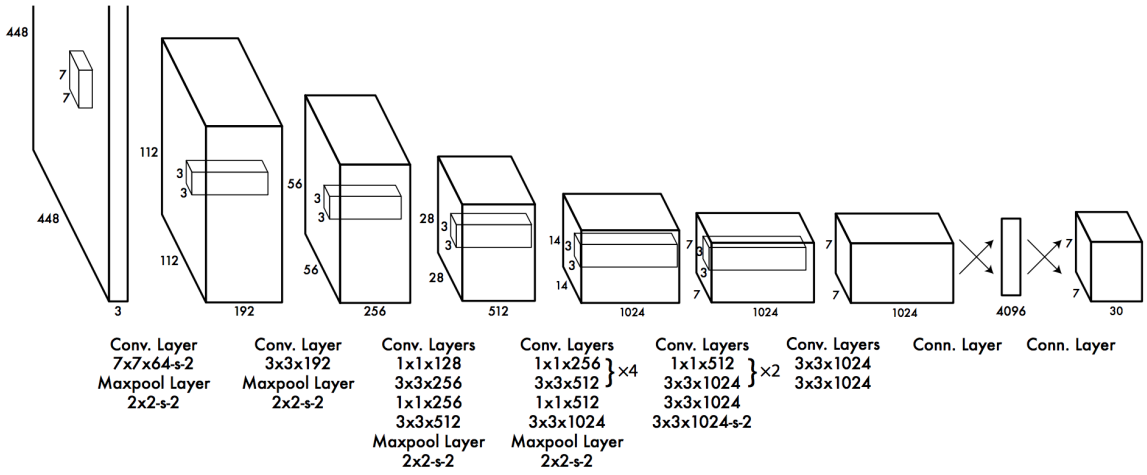

Рисунок 2.5 – Архітектура YOLO [13]

YOLO прогнозує кілька обмежувальних рамок для одного осередку. Щоб підрахувати втрату для істинно позитивного прогнозу, необхідно щоб тільки одна обмежувальна рамка відповідала об'єкту. Для цього, ми вибираємо обмежувальну рамку з найвищим IoU (intersection over union) з правдивої обмежувальної рамкою. Ця стратегія призводить до спеціалізації серед обмежувальних рамок. Кожний прогноз стає кращим в передбаченні обмежувальних рамок певних розмірів і співвідношень сторін.

YOLO використовує суму квадратів помилок між прогнозами і правдивими значеннями для підрахунку втрати.

Функція втрати складається з:

– втрати класифікації;

– втрати локалізації (помилки між спрогнозованої обмежувальної рамкою і правдивої обмежувальної рамкою);

– втрати впевненості (об'єктності обмежувальної рамки).

Остаточна функція втрати, яка наведена нижче, підсумовує помилки локалізації, впевненості і класифікації.

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

YOLO може знаходити обмежувальні рамки, які дублюють один одного. Щоб виправити це, YOLO застосовує алгоритм немаксимального придушення (non-maximum suppression) щоб прибрати повторювані обмежувальні рамки з низькою оцінкою впевненості.

Нижче наведено приклад одного з варіантів реалізації такого алгоритму:

– впорядкувати прогнози за оцінкою впевненості;

– починаючи з високих оцінок, поточні прогнози ігноруються, якщо знаходяться попередні прогнози з цим же класом і IoU > 0.5 в порівнянні з поточним прогнозом;

– повторювати попередній крок, поки всі прогнози не будуть перевірені.

Переваги YOLO:

– швидкість роботи. YOLO добре підходить для обробки зображень в реальному часі;

– прогнози робляться однією нейронною мережею. Вся модель тренується разом.

– YOLO більш генералізована. Ця модель працює краще інших методів при застосуванні на інших типах зображень, наприклад на картинах;

– методи пропозиції регіонів обмежують класифікатор в конкретному регіоні. YOLO має доступ до цілого зображення під час прогнозу меж об’єктів. Використовуючи додатковий контекст, YOLO демонструє менше помилкових позитивів в фонових ділянках зображення;

– YOLO розпізнає один об’єкт в кожній клітині. Це забезпечує просторове різноманіття під час виконання прогнозів.

YOLOv2 це друга версія моделі YOLO, мета якої – значно поліпшити точність, і в той же час зробити модель швидше. [14, 15] В YOLOv2 з'явилися такі покращення:

– У згорткові шари була додана нормалізація батчів;

– YOLOv2 починає тренувати класифікатор із зображеннями з роздільною здатністю 224×224, але потім переналаштовує класифікатор знову з зображеннями з роздільною здатністю 448×448, використовуючи набагато менше епох;

– перша версія YOLO використовувала випадкові здогадки щодо обмежувальних рамок. У реальних проблемах, обмежувальні рамки не є випадковими. Наприклад, автомобілі мають дуже схожі форми, а обмежувальні рамки навколо пішоходів мають приблизне співвідношення сторін 0.41. Початкова тренування стає більш стабільним, якщо обирати обмежувальні рамки, форми яких поширені для об'єктів з реального життя. Наприклад, перед тренуванням можна задати 5 якірних рамок із заздалегідь заданими формами. Тепер, замість передбачення 5 випадкових обмежувальних рамок, модель визначає зміщення щодо кожної з заздалегідь заданих якірних рамок. Якщо обмежити значення зміщень, можна підтримувати різноманітність прогнозів, і кожен прогноз буде фокусуватися на конкретній формі. Через це початкова тренування моделі буде більш стабільним;

– YOLOv2 може приймати на вхід зображення різних розмірів. Якщо ширина і висота вхідного зображення подвоєна, необхідно лише зробити вихідну сітку в 4

рази більше, і отже, необхідно зробити в 4 рази більше прогнозів. Оскільки нейронна мережа YOLO знижує розмірність вхідних даних в 32 рази, необхідно переконатися, що ширина і висота вхідного зображення кратні 32.

– під час тренування, YOLOv2 бере зображення розміром 320×320, 352×352, …, і 608×608 (з кроком 32). Кожні 10 батчів, YOLOv2 випадково вибирає наступний розмір зображення для тренування моделі. Це служить аугментацією даних і змушує мережу робити хороші прогнози для вхідних зображень різного розміру і масштабу.

YOLOv3 – це останнє удосконалення моделі YOLO.

YOLOv3 використовує багатокласову класифікацію і замінює активаційну функцію softmax незалежними логістичними класифікаторами, щоб розрахувати ймовірність приналежності вводу конкретними класами. [16, 17] Замість використання методу середньоквадратичної помилки при підрахунку помилки класифікації, YOLOv3 використовує binary cross-entropy loss для кожного класу. Відмова від використання функції softmax також знижує обчислювальну складність.

У YOLOv3 використовується нова нейронна мережа Darknet-53 замість Darknet-19 в якості мережі для вилучення ознак. Darknet-53 в основному складається з фільтрів розмірністю 3×3 і 1×1 з пропусками з'єднань на зразок залишкових мереж в ResNet (див. рис. 2.6).

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Рисунок 2.6 – Архітектура мережі Darknet-53 [13]

Модель YOLOv3 працює дуже добре в категорії швидких детекторів, коли важлива швидкість обробки зображень.

2.3 Модель MTCNN

MTCNN – нейромережева модель, створена спеціально для знаходження обличчя на цифровому зображенні. Вона містить три досить простих згорткових нейронних мережі, які називають P-Net, R-Net і O-Net. Крім визначення координат обмежувальної рамки навколо знайденого обличчя, ця модель також визначає координати особливих точок, а саме очей, носа і рота. [18, 19] Нижче наведена схема архітектури наведених вище нейронних мереж (див. рис. 2.7).

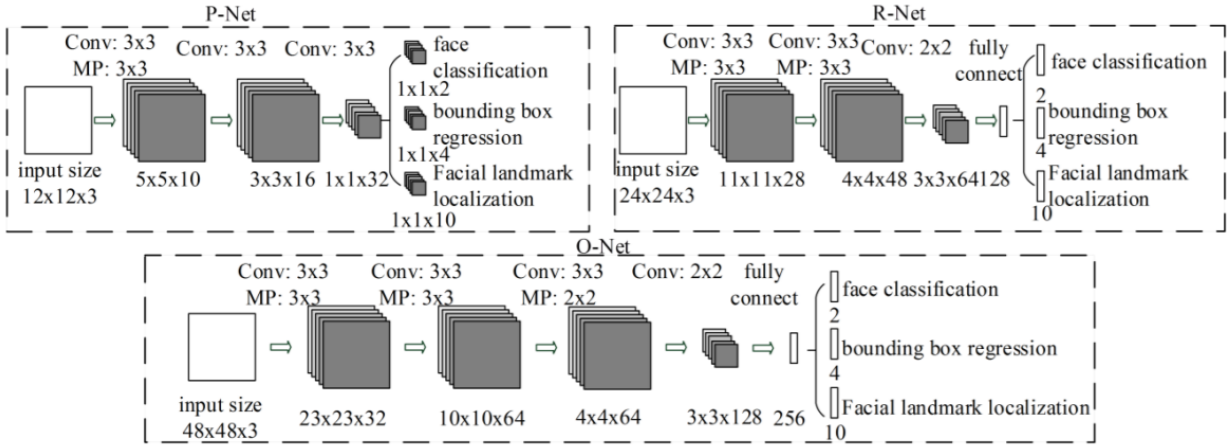

Рисунок 2.7 – Архітектура нейронних мереж MTCNN [18]

Модель MTCNN працює в три стадії.

Спочатку потрібно передати зображення в модель. У MTCNN створюється піраміда зображень, щоб модель могла знаходити обличчя різного розміру. Іншими словами, модель використовує кілька копій зображення різного розміру.

Для кожної копії, ядро розміром 12 на 12 пікселів проходить по всьому зображенню із зсувом 2. Це означає, що на першому етапі верхній лівий кут ядра

знаходиться в координатах (0,0), а нижній правий кут ядра – (12,12). На другому етапі, верхній лівий кут ядра буде знаходиться в координатах (2,0), а нижній правий – (14,12). Частина зображення, на якій знаходиться ядро, обробляється першої нейронною мережею, P-Net, яка повертає координати обмежувальної рамки, якщо знаходить на частині зображення обличчя.

Кожне ядро буде менше відносно великого зображення, тому воно зможе знайти маленькі обличчя на великих зображеннях. У той же час, ядро буде більше відносно маленького зображення, через це воно зможе знайти великі обличчя на зображенні меншого масштабу.

Після передачі зображення в модель, вона створює кілька масштабованих копій зображення і передає їх в першу нейронну мережу – P-Net – і збирає її вивід.

P-Net навчена так, що вона виводить відносно точну обмежувальну рамку для кожного ядра. Проте, мережа більш впевнена в деяких обмежувальних рамках порівняно з іншими. Таким чином, необхідно проаналізувати вихідні дані P-Net і отримати список рівнів впевненості для кожної обмежувальної рамки, і видалити рамки з низьким рівнем впевненості (тобто рамки, в яких мережа не впевнена, що в них знаходиться особа).

Після того, як будуть відібрані прямокутники з більшим ступенем впевненості, необхідно стандартизувати систему координат, перетворивши всі координати в систему координат реального, немасштабованого зображення. Оскільки більшість ядер знаходяться в зменшеному зображення, їх координати будуть засновані на меншому зображенні.

Однак залишається ще багато обмежувальних рамок, багато з яких перекривають один одного. Алгоритм немаксимального придушення, (Non-Maximum Suppression, NMS) застосовується для зменшення кількості обмежувальних рамок.

У MTCNN, NMS проводиться спочатку шляхом сортування обмежувальних рамок (і їх відповідних ядер) по їх рівню впевненості, або оцінки. У деяких інших моделях NMS вибирає найбільшу обмежувальну рамку замість тієї, в якій мережа найбільш впевнена.

Потім обчислюється площа кожного з ядер, а також площа перекриття між кожним ядром і ядром з найбільшою оцінкою. Обмежувальні рамки ядер, які сильно перекриваються з обмежувальною рамкою ядра з найвищою оцінкою, видаляються. Після цього NMS повертає список обмежувальних рамок, що залишилися.

NMS застосовується один раз для кожного масштабованого зображення, потім ще один раз з усіма залишаються ядрами кожного масштабу. Це позбавляє від зайвих обмежувальних рамок, дозволяючи звузити пошук до одного точного прямокутника на кожне обличчя.

Після цього координати прямокутника конвертуються в координати реального зображення. Зараз координати кожного прямокутника мають значення від 0 до 1. Примножуючи координати на ширину і висоту початкового зображення, ми можемо перетворити координати обмежувальної рамки в координати початкового зображення.

Оскільки обмежувальні прямокутники можуть бути не квадратними, вони перетворюються в квадрат, шляхом збільшення більш коротких сторін (якщо ширина менше висоти, прямокутник розширюється в сторони; якщо висота менше ширини, прямокутник розширюється вертикально).

Після цього отримані координати обмежувальних рамок передаються на другу стадію.

Іноді зображення може містити тільки частину обличчя, в той час як решта особа перебуває за межами зображення. У цьому випадку мережа може повернути обмежувальну рамку, яка частково виходить за межі зображення.

Для кожного обмежувального прямокутника створюється масив відповідного розміру, і значення пікселів копіюються в новий масив. Якщо обмежувальний прямокутник виходить за межі зображення, копіюється тільки частина зображення в прямокутнику, яка знаходиться в межах зображення, і елементи масиву, що залишилися заповнюються нулями. Цей процес заповнення масивів нулями називається padding (заповнення).

Після заповнення масивів обмежувальних прямокутників, їх розмір змінюється до 24 x 24 пікселів, а їх значення нормалізуються таким чином, щоб вони набували значення від -1 до 1.

Тепер, коли є велика кількість зображень розміром 24 x 24 (стільки ж, скільки число обмежувальних прямокутників, які залишилися після першого етапу), зображення передаються в наступну нейронну мережу, R-Net.

Вивід R-Net аналогічний виводу P-Net: він включає в себе координати нових, більш точних обмежувальних рамок, а також рівень впевненості кожної з цих обмежувальних рамок.

Після цього ще раз обмежувальні з меншим рівнем впевненості відкидаються, і застосовується алгоритм NMS для подальшого усунення надлишкових обмежувальних рамок. Оскільки координати цих нових обмежувальних прямокутників засновані на обмежувальних прямокутниках, які були отримані за допомогою P-Net, їх необхідно перетворити в стандартні координати.

Після стандартизації координат, обмежувальні рамки перетворюються в квадрат для їх передачі в останню нейронну мережу, O-Net.

Обмежувальні рамки, які виходять за межі зображення, заповнюються нулями. Потім розмір всіх обмежувальних рамок, отриманих за допомогою O-Net, змінюється до 48 x 48 пікселів. Після цього обмежувальні рамки передаються в O-Net.

Вивід O-Net трохи відрізняється від виводів P-Net і R-Net. O-Net надає 3 виводи: координати обмежувальної рамки, координати 5 особливих точок, і рівень впевненості кожної рамки.

Ще раз, прямокутники з низьким рівнем впевненості відкидаються. Координати прямокутника і координати особливих точок стандартизуються. Нарешті, останній раз застосовується алгоритм NMS. На цьому етапі для кожного обличчя на зображенні повинна залишитися тільки одна обмежувальна рамка.

Модель MTCNN має багато переваг. Низька складність обчислень призводить до швидкої роботі моделі. Для досягнення достатньої продуктивності

для роботи в режимі реального часу модель переміщує ядро із зсувом, рівним 2, скорочуючи кількість операцій до чверті. Модель також починає шукати особливі точки тільки в останній мережі (O-Net), після того як координати обмежувального прямокутника були відкалібровані, що звужує область, в якій модель шукає особливі точки. Це також робить модель більш швидкою і ефективною.

Висока точність досягається за допомогою застосування глибоких нейронних мереж. Наявність трьох мереж – кожна з декількома шарами – забезпечує більш високу точність, оскільки кожна мережа може точніше калібрувати результати попередньої. Крім того, ця модель використовує піраміду зображень, щоб знаходити як і великі, так і маленькі. Хоч це може привести до великої кількості знайдених обмежувальних рамок, алгоритм NMS, а також R-Net і O-Net, допомагають відхилити велику кількість помилкових обмежувальних рамок.

2.4 Формулювання гіпотез

Після аналізу досліджуваних алгоритмів, були визначені наступні гіпотези:

– модель YOLO працює значно швидше моделі MTCNN, оскільки модель YOLO містить одну нейронну мережу, в той час як модель MTCNN використовує три нейронних мережі, які застосовуються багато разів під час обробки одного зображення;

– ефективність обох моделей на зображеннях, на яких особи не перекриті і добре видимі, є однаково високою;

– модель MTCNN на зображеннях, на яких обличчя перекриті або частково перебувають за межами кадру, є більш ефективною, ніж модель YOLO, оскільки модель MTCNN створювалась для вирішення задачі знаходження облич на зображеннях, в той час як модель YOLO створювалась для знаходження будь-яких об’єктів на зображеннях;

– незважаючи на те, що модель MTCNN працює краще моделі YOLO на зображеннях, на яких обличчя частково перекриті або частково перебувають за межами кадру, ефективність моделі MTCNN для таких завдань є досить низькою.

Наведені вище гіпотези будуть перевірені за допомогою проведення експерименту, подробиці проведення якого описані в наступній частині роботи.

3 ОПИС І АНАЛІЗ РЕЗУЛЬТАТІВ ЕСПЕРІМЕНТУ

Для порівняння ефективності різних алгоритмів необхідно визначиться з чисельної метрикою, яка відображає ефективність алгоритмів. Для алгоритмів знаходження об'єктів на зображенні, цією метрикою є середня точність (average precision). [20]

Ця метрика використовує Intersection Over Union, точність (precision), і повноту (recall). [21]

Intersection Over Union (IOU) – це показник, заснований на коефіцієнті Жакарра, який оцінює ступінь перекриття між двома обмежувальними прямокутниками. Для цього потрібна наявність правдивого обмежувального прямокутника і прогнозованого обмежувального прямокутника. Застосовуючи IOU, можна визначити, чи є виявлення об'єкта вірним (True Positive) або хибним (False Positive).

IOU розраховується як співвідношення площі перетину правдивого обмежувального прямокутника і прогнозованого обмежувального прямокутника і площі об'єднання правдивого обмежувального прямокутника і прогнозованого обмежувального прямокутника за наступною формулою:

$$IOU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{3.1}$$

Зображення нижче ілюструє IOU між правдивою обмежувальної рамкою (зеленої) і прогнозованою обмежувальної рамкою (червоною) (див. рис. 3.1).

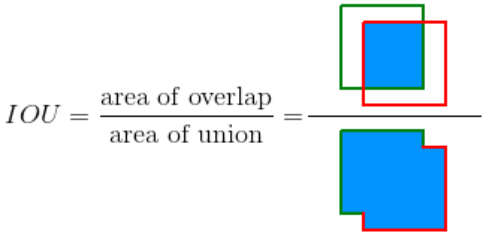

Рисунок 3.1 – Приклад розрахування IOU

Метрики точності і повноти використовують вірно позитивний прогноз, помилково позитивний прогноз і помилково негативний прогноз.

Вірно позитивний прогноз (True Positive, TP) означає правильне розпізнавання об’єкта, коли IOU більше заданого порогу.

Помилково позитивний прогноз (False Positive, FP) означає невірне розпізнавання об’єкта, коли IOU менше заданого порогу.

Помилково негативний прогноз (False Negative, FN) означає, що модель не виявила об’єкт, який знаходиться на зображенні.

Поріг зазвичай задається значеннями 0.5, 0.75 і 0.95.

Точність – це здатність моделі ідентифікувати тільки актуальні об’єкти. Вона дорівнює відсотку вірно позитивних прогнозів і визначається наступною формулою:

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{all\ detections} \tag{3.2}$$

Повнота – це здатність моделі знаходити всі актуальні об’єкти (все правдиві обмежувальні рамки). Це співвідношення кількості вірно позитивних прогнозів і кількості всіх об’єктів, які повинна виявити модель, яке визначається як

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{all\ ground\ truths} \tag{3.3}$$

Для підрахунку середньої точності спочатку необхідно побудувати криву точності і повноти (precision x recall curve).

Крива точності і повноти – це хороший спосіб оцінити ефективність детектора об’єктів, шляхом побудови кривої для кожного класу об’єктів по мірі зменшення оцінки впевненості. Детектор об’єктів певного класу вважається хорошим, якщо його точність залишається високою при збільшенні повноти, що означає, що при зміні порога впевненості, точність і повнота все одно залишатимуться високими. Ще один спосіб визначення ефективного детектора

об’єктів – знаходження детектора, який може ідентифікувати тільки актуальні об’єкти (0 помилково позитивних прогнозів, висока точність), знаходячи всі присутні об’єкти (0 помилково негативних прогнозів, висока повнота).

Поганий детектор об'єктів повинен збільшити кількість виявлених об’єктів (збільшення помилкових позитивних прогнозів, більш низька точність), щоб знайти всі присутні об’єкти (висока повнота). Ось чому крива точності і повноти зазвичай починається з високих значень точності, які зменшуються по мірі збільшення повноти.

Іншим способом порівняння ефективності детекторів об’єктів є обчислення площі під кривою точності і повноти (AUC, area under curve). Оскільки криві часто є звивистими, що йдуть вгору і вниз, порівняння різних кривих (різних детекторів) на одному і тому ж графіку зазвичай є непростим завданням, оскільки криві часто перетинаються одна з одною. Ось чому середня точність (AP, average precision), яка є числовий метрикою, допомагає в порівнянні різних детекторів. На практиці середня точність - це точність, усереднена за всіма значеннями повноти від 0 до 1.

Загальне визначення середньої точності (AP) - це знаходження області під кривою точності і повноти знаходиться за наступною формулою

$$AP = \int_0^1 p(r)dr \tag{3.4}$$

де r – повнота;

p – точність.

Таким чином, використовуючи середню точність, яка приймає значення від 0 до 1, можна порівнювати ефективність різних моделей-детекторів об'єктів.

Для порівняння двох моделей, YOLO і MTCNN, в експерименті використовувався набір даних Wider Face. Набір даних WIDER FACE – це набір даних для алгоритмів знаходження облич, зображення в якому були обрані з загальнодоступного набору даних WIDER. Були відібрані 32 203 зображення, що містять 393 703 обличчя з високим ступенем мінливості в масштабі, позі і оклюзії [22].

Для порівняння були обрані реалізації моделей з відкритим похідним кодом YOLOv3 [23] і MTCNN [24], які використовують фреймворк для глибокого навчання TensorFlow [25] і бібліотеку Keras. Модель YOLO була заздалегідь навчена знаходити різні об'єкти на зображеннях. В процесі виконання експерименту ця модель була спеціально навчена знаходити особи, використовуючи набір даних Wider Face. Оскільки модель MTCNN була спочатку створена для знаходження облич, необхідності додаткового навчання моделі в процесі виконання експерименту не виникло.

Для навчання моделі YOLO використовувалися 10 тисяч зображень. Для оцінки ефективності моделей використовувалися дві з половиною тисячі зображень.

Нижче наведені результати експерименту (див. табл. 3.1).

Таблиця 3.1 – результати експерименту

	YOLO	MTCNN
Тренувальний набір даних, 10,000 зображень, середня точність	32.55%	41.71%
Тестовий набір даних, 2,500 зображень, середня точність	30.88%	38.35%

Нижче наведені графіки з кривими точності і повноти (див. рис. 3.2 – 3.5).

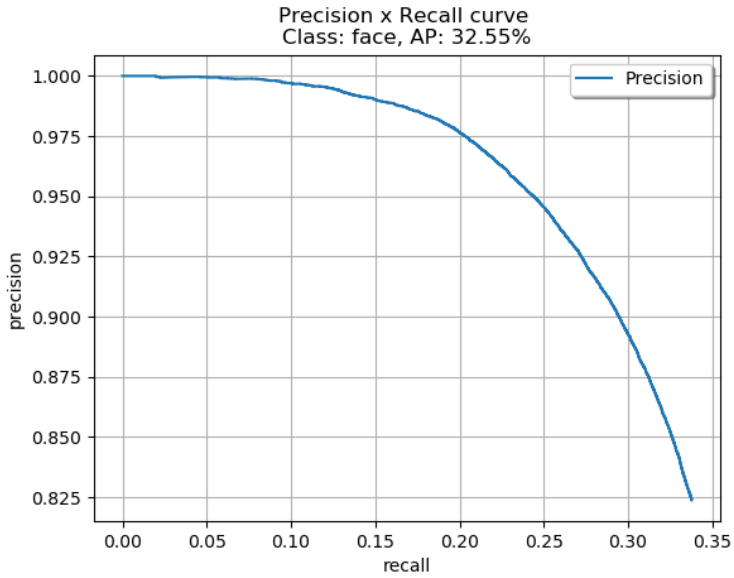

Рисунок 3.2 – Крива точності і повноти для тренувального набору даних і моделі YOLO

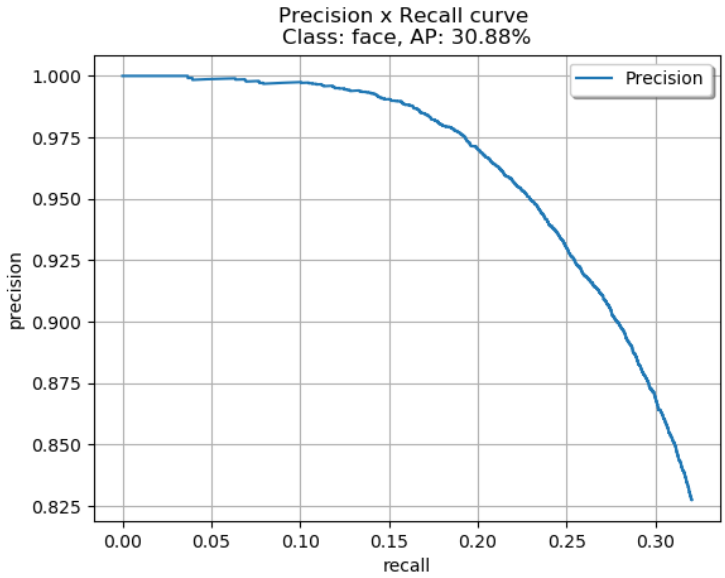

Рисунок 3.3 – Крива точності і повноти для тестового набору даних і моделі YOLO

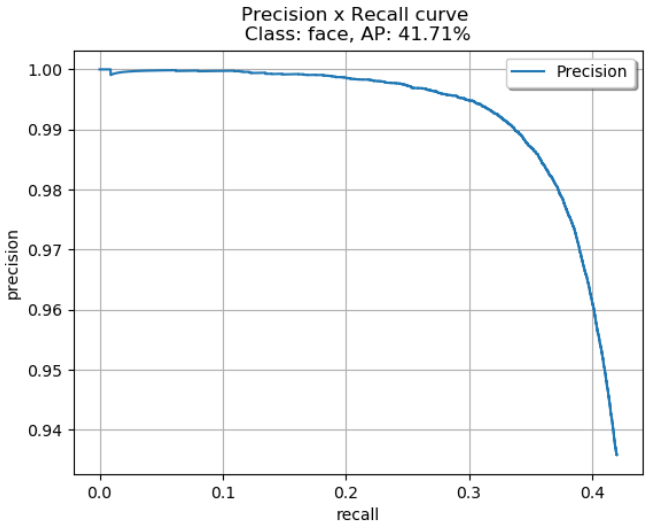

Рисунок 3.4 – Крива точності і повноти для тренувального набору даних і моделі MTCNN

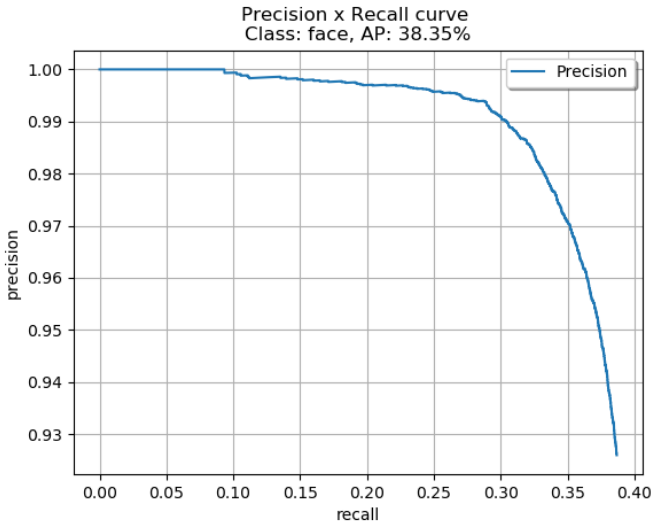

Рисунок 3.5 – Крива точності і повноти для тестового набору даних і моделі MTCNN

Проаналізувавши наведені вище дані, можна зрозуміти, що модель MTCNN ефективніше моделі YOLO в контексті завдання знаходження облич. Наприклад, в тестовому наборі даних, модель YOLO знайшла 32% всіх облич, і 83% всіх виявлень, які зробила модель, є вірними. У той же час, модель MTCNN знайшла 39% всіх облич, і при цьому 93% всіх виявлень виявилися вірними.

Нижче наведена таблиця з часом роботи моделей (див. табл. 3.2).

Таблиця 3.2 – Час роботи моделей

	YOLO	MTCNN
Час роботи на тренувальному наборі даних, секунди	978.45	2475
Час роботи на тестовому наборі даних, секунди	269.77	732.59

Модель YOLO працює значно швидше моделі MTCNN, тому що YOLO використовує одну нейронну мережу, в той час як MTCNN використовує 3 нейронних мережі і застосовує їх багато разів за обробку одного зображення.

Наступний етап експерименту - перевірити, наскільки добре моделі знаходять обличчя, що не перекриті і які добре видно. Результати цього експерименту наведені нижче (див. табл. 3.3).

Таблиця 3.3 – Порівняння ефективності моделей на зображеннях, на яких обличчя не перекриті і цілком видні

	YOLO	MTCNN
Середня точність	86.52%	84.4%

Нижче наведені графіки з кривими точності і повноти для набора даних с зображеннями, на яких обличчя не перекриті і цілком видні (див. рис. 3.6 – 3.7).

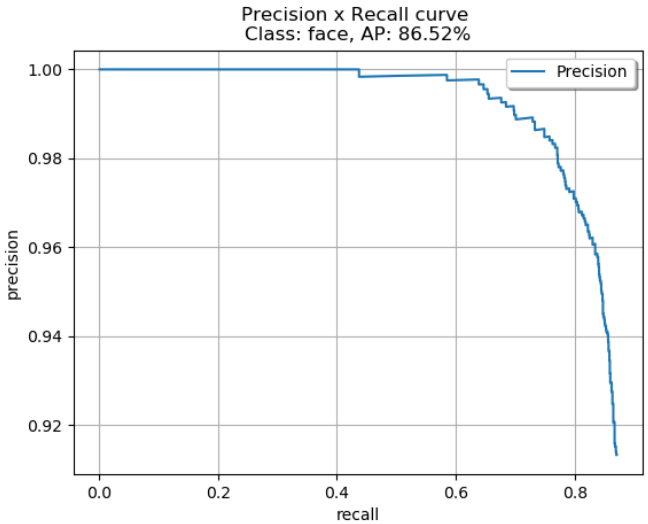

Рисунок 3.6 – Крива точності і повноти для набору даних, де обличчя цілком видно, і моделі YOLO

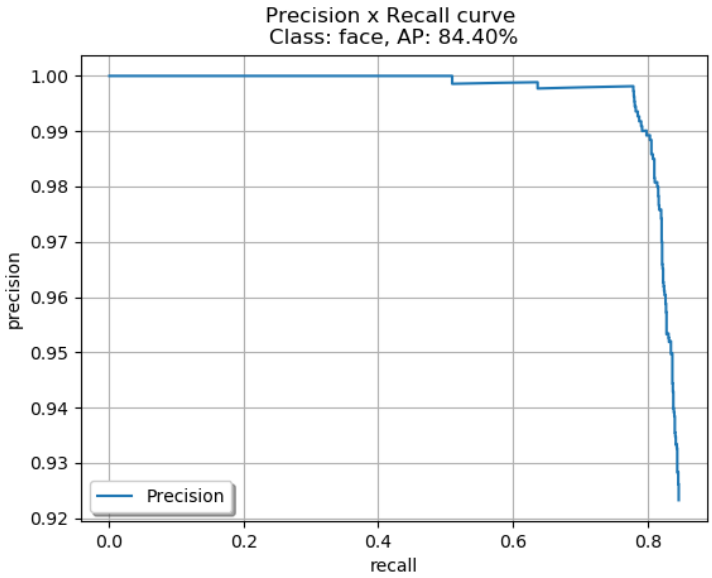

Рисунок 3.7 – Крива точності і повноти для набору даних, де обличчя цілком видно, і моделі MTCNN

В таких умовах, модель YOLO показала себе трохи краще, ніж модель MTCNN. YOLO виявила 87% всіх облич на зображеннях, і 91% виявлень є вірними. MTCNN виявила 85% всіх облич, і 92% всіх виявлень виявилися вірними.

Можна зробити висновок, що якщо на вхідних зображеннях більшість облич нічим не перекриті і їх добре видно, і немає необхідності виявляти особливі точки, то для таких завдань слід використовувати модель YOLO, оскільки вона значно швидше моделі MTCNN і навіть трохи перевершує її за ефективністю.

У наступному експерименті перевіряється, наскільки ефективно обидві моделі обробляють зображення, де більше половини облич перекриті різними об'єктами або частково перебувають за межами кадру. Кількість зображень в наборі даних для цього експерименту становило 377 зображень. Результати цього експерименту наведені нижче (див. табл. 3.4).

Таблиця 3.4 – Порівняння ефективності моделей на зображеннях, на яких більше половини облич перекриті або частково перебувають за межами кадру

	YOLO	MTCNN
Середня точність	9.73%	17.1%

Нижче наведені графіки з кривими точності і повноти для набора даних с зображеннями, на яких більше половини облич перекриті або частково перебувають за межами кадру (див. рис. 3.8 – 3.9).

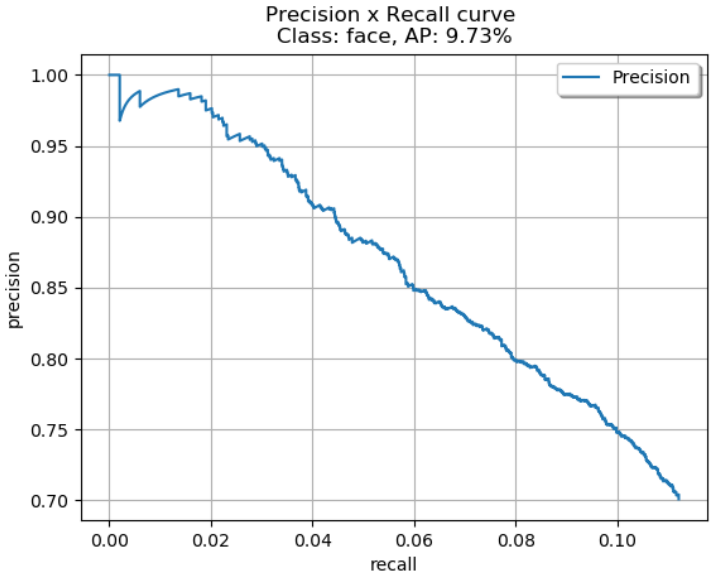

Рисунок 3.8 – Крива точності і повноти для набору даних, в якому більше половини облич на зображені перекриті або частково перебувають за межами кадру, і моделі YOLO

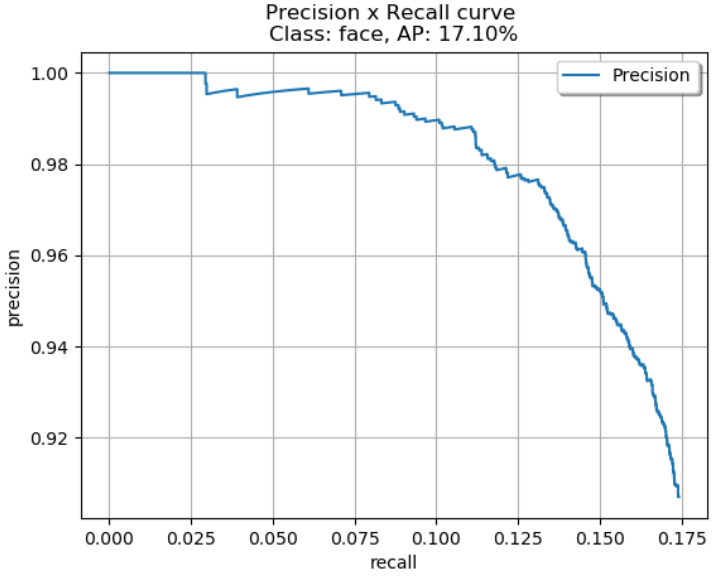

Рисунок 3.9 – Крива точності і повноти для набору даних, в якому більше половини облич на зображені перекриті або частково перебувають за межами кадру, і моделі MTCNN

На основі результатів експерименту, можна зробити висновок, що в неоптимальних умовах модель MTCNN працює набагато ефективніше моделі YOLO. Модель MTCNN виявила 17% всіх облич, при цьому 91% всіх виявлень виявилися правильними. При цьому модель YOLO виявила лише 11% всіх облич, і лише 70% всіх виявлень виявилися правильними.

В результаті проведення експерименту гіпотези, сформульовані в попередній частині роботи, підтвердилися. Підводячи підсумок експерименту, можна зробити наступні висновки:

– модель YOLO працює приблизно в два з половиною рази швидше моделі MTCNN;

– в оптимальних умовах, ефективність двох моделей є однаково високою, хоча модель YOLO демонструє трохи більшу повноту, ніж модель MTCNN;

– в неоптимальних умовах, модель MTCNN працює набагато ефективніше моделі YOLO, хоча варто зауважити, що обидві моделі демонструють недостатню ефективність для вирішення подібних завдань;

– в змішаних умовах, модель MTCNN має невелику перевагу над моделлю YOLO.

На основі зроблених вище висновків, можна визначити наступні рекомендації по використанню моделей YOLO і MTCNN для знаходження облич на зображеннях:

– якщо крім осіб потрібно знаходити особливі точки, слід використовувати модель MTCNN, оскільки модель YOLO не знаходить особливі точки на знайдених обличчях;

– якщо обсяг наявних обчислювальних ресурсів обмежений, слід використовувати модель MTCNN, тому що хоч в моделі YOLO й використовується одна нейронна мережа, її архітектура набагато складніше нейронних мереж моделі MTCNN, і вона вимагає більшого обсягу обчислювальних ресурсів для роботи;

– якщо важлива швидкість роботи моделі, слід використовувати модель YOLO, оскільки модель MTCNN працює значно повільніше моделі YOLO;

– якщо на вхідних зображеннях обличчя добре видно і вони не перекриті, слід використовувати модель YOLO, оскільки в оптимальних умовах моделі працюють однаково ефективно, але модель YOLO працює значно швидше.

Для задач, коли необхідно виявляти особи, які перекриті або частково перебувають за межами кадру, пропонується використовувати комбінацію двох моделей.

Проведений експеримент показав, що в неоптимальних умовах модель MTCNN працює значно краще моделі YOLO, але недостатньо добре для вирішення подібних завдань. Разом з цим модель YOLO дозволяє виявляти будь-які об’єкти на зображенні.

Запропонований метод працює в 4 етапи:

– на першому етапі, за допомогою моделі YOLO знаходяться обмежувальні рамки навколо фігур людей, присутніх на зображенні;

– на другому етапі, знайдені обмежувальні рамки розширюються на заздалегідь визначену кількість пікселів, щоб компенсувати можливу похибку при визначенні координат обмежувальної рамки. Зображення, що знаходяться всередині рамки вирізаються і передаються в якості введення в модель MTCNN;

– на третьому етапі, модель MTCNN визначає обличчя людини на вирізаної частини зображення;

– на четвертому етапі, координати осіб, отримані від MTCNN, трансформуються в систему координат вхідного зображення, і застосовується алгоритм NMS для виключення повторюваних обмежувальних рамок.

Основною перевагою цього методу є те, що модель MTCNN знаходить обличчя не на всьому зображенні, а на частині зображення, де вже точно відомо, що знаходиться людина. Піраміда зображень, яку будує MTCNN, будується для частини зображення, через що розмір обличчя під час обробки цієї частини зображення виходить більшим, ніж якби MTCNN обробляла все зображення цілком. Це дозволяє моделі MTCNN ефективніше знаходити обличчя меншого розміру, обличчя, які є частково перекритими, та обличчя, які частково перебувають за межами кадру.

Основним недоліком цього методу є те, що цей метод працює досить повільно, оскільки модель MTCNN застосовується для кожного регіону, де знаходиться фігура людини. Це робить роботу моделі дуже повільною для зображень, на яких знаходяться багато людей.

ВИСНОВКИ

В процесі виконання роботи був проведений експеримент, в результаті якого були визначені рекомендації по використанню досліджуваних моделей. Через високу обчислювальну складність, модель Faster R-CNN не розглядалась в експерименті, хоча вона є хоч і повільним, але ефективним детектором об’єктів.

Модель YOLO працює трохи ефективніше моделі MTCNN в оптимальних умовах, тобто на зображеннях з добре видимими особами. Модель YOLO також працює набагато швидше моделі MTCNN. Модель MTCNN є значно більш ефективною моделлю ніж модель YOLO в неоптимальних умовах, тобто на зображеннях з частково перекритими особами або особами, які частково знаходяться за межами кадру. Але незважаючи на те, що в цих умовах модель MTCNN значно ефективніше моделі YOLO, ефективність цієї моделі недостатньо висока для таких завдань.

Модель YOLO потрібно застосовувати в задачах, де важлива швидкість роботи моделі, або де обличчя на вхідних зображеннях не є перекритими іншими об'єктами та досить великі відносно вхідного зображення. Також модель YOLO слід використовувати, якщо крім облич необхідно знаходити інші об’єкти. Модель MTCNN слід застосовувати у випадках, якщо обличчя на вхідних зображеннях частково перекриті або частково перебувають за межами кадру. Також модель MTCCN необхідно застосовувати, якщо необхідно разом з обличчям знаходити особливі точки.

Також в результаті роботи був запропонований метод знаходження перекритих осіб або осіб невеликого розміру щодо вхідного зображення, заснований на комбінації моделей YOLO і MTCNN. Модель YOLO спочатку знаходить на зображенні область, в якій знаходиться людина, і модель MTCNN в цій області знаходить обличчя. Локалізація області пошуку дозволяє підвищити ефективність знаходження облич невеликого розміру і облич, перекритих іншими об’єктами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Smelyakov, K., Tovchyrechko, D., Ruban, I., Chupryna, A., Ponomarenko, O. Local feature detectors performance analysis on digital image // 2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019.

2. Smelyakov, K., Hvozdiev, M., Chupryna, A., Sandrkin, D., Martovytskyi, V. Comparative efficiency analysis of gradational correction models of highly lighted image // 2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019.

3. Smelyakov, K., Smelyakov, S., Chupryna, A. Adaptive edge detection models and algorithms // Studies in Computational Intelligence, 2020.

4. Smelyakov, K., Datsenko, A., Skrypka, V., Akhundov, A. The efficiency of images reduction algorithms with small-sized and linear details // 2019 IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology, PIC S and T 2019.

5. Smelyakov, K., Shupyliuk, M., Martovytskyi, V., Tovchyrechko, D., Ponomarenko, O. Efficiency of image convolution // Proceedings of the International Conference on Advanced Optoelectronics and Lasers, CAOL, 2019.

6. Asit Kumar Datta, Madhura Datta, Pradipta Kumar Banerjee Face Detection and Recognition. – CRC Press, 2015. – 352 c.

7. G. Blokdyk Facial Recognition A Complete Guide. – 5STARCooks, 2018. – 126 c.

8. Girshick, R., Donahue, J., Darrell, T., Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation // 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014.

9. How R-CNN Works on Object Detection? // URL: https://medium.com/towards-artificial-intelligence/how-r-cnn-works-on-object-detection-443679b0187c (дата звернення: 10.05.2020).

10. Girshick, R. Fast R-CNN // 15th IEEE International Conference on Computer Vision, ICCV 2015.

11. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks // 29th Annual Conference on Neural Information Processing Systems, NIPS 2015.

12. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms // URL: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (дата звернення: 10.05.2020).

13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You only look once: Unified, real-time object detection // 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016.

14. Redmon, J., Farhadi, A. YOLO9000: Better, faster, stronger // 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017.

15. YOLO – You only look once, real time object detection explained // URL: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006 (дата звернення: 10.05.2020).

16. Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement // arXiv:1804.02767, 2018.

17. What’s new in YOLO v3? // URL: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b (дата звернення: 10.05.2020).

18. Zhang, K., Zhang, Z., Li, Z., Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks // IEEE Signal Processing Letters, 2016.

19. How Does A Face Detection Program Work? (Using Neural Networks) // URL: https://towardsdatascience.com/how-does-a-face-detection-program-work-using-neural-networks-17896df8e6ff (дата звернення: 10.05.2020).

20. mAP (mean Average Precision) for Object Detection // URL: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173 (дата звернення: 10.05.2020).

21. Most popular metrics used to evaluate object detection algorithms // URL: https://github.com/rafaelpadilla/Object-Detection-Metrics (дата звернення: 10.05.2020).

22. Yang, S., Luo, P., Loy, C.C., Tang, X. WIDER FACE: A face detection benchmark // 29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016.

23. A Keras implementation of YOLOv3 (Tensorflow backend) // URL: https://github.com/qqwweee/keras-yolo3 (дата звернення: 10.05.2020).

24. MTCNN face detection implementation for TensorFlow // URL: https://github.com/ipazc/mtcnn (дата звернення: 10.05.2020).

25. TensorFlow: An end-to-end open source machine learning platform // URL: https://www.tensorflow.org (дата звернення: 10.05.2020).