

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень


Система розпізнавання зображень на базі методів, заснованих на згортковій
нейромережі
Назва теми

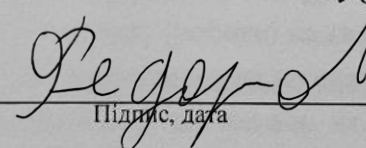
КВРКІ. 200231.20.02.07 ПЗ
Шифр

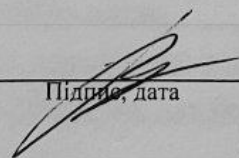
Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

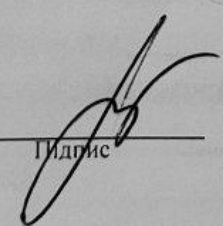
Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група KI2-20-2  Б. В. Гончар
Підпис Ініціали, прізвище

Керівник  Є. Є. Федоров
Підпис, дата Ініціали, прізвище

Нормоконтролер  І.О. Засорнова
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

 Т.О. Говорущенко
Підпис Ініціали, прізвище

«12» червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Гончару Богдану Валентиновичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система розпізнавання зображень на базі методів, заснованих на згортковій нейромережі

Керівник проекту (роботи) Федоров Є.Є., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Сучасні системи згорткових нейромереж

Дослідження методів та аналіз згорткових нейромереж

Використання згорткових нейромереж для безпеки розпізнавання об'єктів у фото та відео потоці

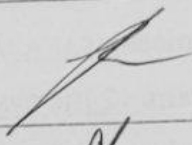



5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

UML Схема програмного забезпечення

Зображення архітектур згорткових нейромереж

Код роботи CNN

6. Консультанти розділів дипломного проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|--|---|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Засорнова І.О., доцент кафедри КПС |  |  |
| Антиплагіат | Нічепорук А.О., доцент кафедри КПС |  |  |

7. Дата видачі завдання

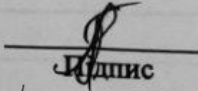
« 10 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

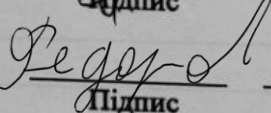
| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1 | Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником | 10.01.2024 | виконано |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2024 | виконано |
| 3 | Робота над розділом 1 – дослідження предметної області та постановка задачі | 01.03.2024 | виконано |
| 4 | Робота над розділом 2 – вибір компонентів для дослідження методів та аналізу згорткових нейромереж | 01.04.2024 | виконано |
| 5 | Робота над розділом 3 – використання згорткових нейромереж для безпеки розпізнавання об'єктів у фото та відео потоці | 29.04.2024 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | 25.05.2024 | виконано |
| 7 | Попередній захист ВКР | 26.05.2024 | виконано |
| 8 | Захист ВКР на засіданні ЕК | Червень 2024 року | |

Студент


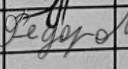

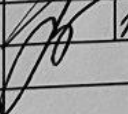
Керівник роботи


Підпис

Б. В. Гончар
Ініціали, прізвище


Підпис

Є. Є. Федоров
Ініціали, прізвище

| № р я д к а | Ф о р м а т | Позначення | Найменування | К і л · л и с т і в | № ек з | П р и м і т к а |
|----------------------------|----------------------------|-------------------------|---|--|---------------|--------------------------------------|
| | | | <u>Текстові документи</u> | | | |
| 1 | | КПКІ 200231.20.02.07 ПЗ | Пояснювальна записка | 69 | | |
| | | | <u>Графічні матеріали</u> | | | |
| 2 | | КПКІ 200231.20.02.07 Е8 | UML Схема програмного забезпечення | 1 | | |
| 3 | | КПКІ 200231.20.02.07 Е8 | Зображення архітектур згортальних нейромереж | 1 | | |
| 4 | | КПКІ 200231.20.02.07 Е8 | Код роботи CNN | 1 | | |
| КвРКІ 200231.20.02.07 ПЗ | | | | | | |
| Зм | Арк | № докум | Підпис | Дата | | |
| Розробив | Гончар | |  | | Літера | Аркуш |
| Перевір. | Федоров | |  | | У | 1 |
| Н. конпр. | Засорнова | |  | | ХНУ, КІ2-20-2 | |
| Затв. | Говорущенко | |  | 12.06 | | |

Система розпізнавання зображень на базі методів, заснованих на згортковій нейромережі

Літера: У, Аркуш: 1, Аркушів: 69

ХНУ, КІ2-20-2

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система розпізнавання зображень на базі методів заснованих на згортковій нейромережі».

Автор роботи: Гончар Богдан Валентинович.

Керівник роботи: Федоров Євген Євгенович.

Пояснювальна записка: 69 с., 50 рис., 3 дод., 52 джерел.

Графічна частина: 3 креслення.

ЗГОРТКОВІ НЕЙРОМЕРЕЖІ, ДОСЛІДЖЕННЯ, МЕТОДИ, АНАЛІЗ, БЕЗПЕКА, РОЗПІЗНАВАННЯ, ВИКОРИСТАННЯ.


Метою роботи є розробка ефективної системи розпізнавання зображень на базі методів, заснованих на згортальній нейромережі.

Об'єктом дослідження є процес розпізнавання зображень

Предметом дослідження є методи розпізнавання зображень на основі згортальній нейромережі.

Завданням дослідження було:

- дослідити та аналізувати згортальні нейромережі та їх методи;
- використати згортальні нейромережі, для безпеки розпізнавання об'єктів у фото та відео потоці;
- зрівняти алгоритми методів згортальних нейромереж.

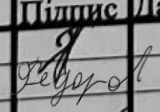



Підпис студента

30.05.2024
Дата

ЗМІСТ

| | |
|---|-----------|
| ВСТУП | 3 |
| 1 СУЧАСНІ СИСТЕМИ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ | 4 |
| 1.1 Опис сучасних згортальних нейромережі | 4 |
| 1.2 Методи роботи сучасних згортальних нейромереж | 7 |
| 1.3 Порівняння згортальних нейромереж | 16 |
| 1.4 Висновки | 21 |
| 2 ДОСЛІДЖЕННЯ МЕТОДІВ ТА АНАЛІЗ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ | 22 |
| 2.1 Початок роботи згортальних нейромереж | 22 |
| 2.2 Згорткові шари та їх робота | 26 |
| 2.3 Реалізація TensorFlow | 32 |
| 2.4 Вимоги до пам'яті та об'єднання шарів | 37 |
| 2.5 Архітектури CNN | 41 |
| 2.6 Висновки | 44 |
| 3 ВИКОРИСТАННЯ ЗГОРТКОВИХ НЕЙРОМЕРЕЖ ДЛЯ БЕЗПЕКИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ФОТО ТА ВІДЕО ПОТОЦІ | 45 |
| 3.1 Розпізнавання образів та їх робота | 45 |
| 3.2 Плюси і мінуси методів обробки зображень та глибокого навчання | 46 |
| 3.3 Використання розпізнавання образів для застосування та їх обробки . | 46 |
| 3.4 Методи розпізнавання образів та їх робота | 49 |
| 3.5 Зрівняння алгоритмів розпізнавання образів | 65 |
| 3.6 Висновки та перспективи | 68 |
| ВИСНОВОК | 69 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ | 70 |
| Додаток А | 76 |
| Додаток Б | 77 |
| Додаток В | 78 |

КвРКІ 200231.20.02.07 ПЗ

| Зм. | Арк. | №докум. | Підпис | Дата | | Літера | Аркуш | Аркушів |
|----------|------|------------------|---|-------|--|---------------|-------|---------|
| Виконав | | Гончар Б.В. |  | | Система розпізнавання зображень на базі методів заснованих на згортковій нейромережі | у | | |
| Перевір. | | Федоров Є.Є. |  | | | | 2 | 69 |
| Н.контр. | | Засорнова І.О. |  | 12.06 | | ХНУ КІ-2-20-2 | | |
| Затвер. | | Говорущенко Т.О. | | | | | | |

ВСТУП

У теперішній час люди все більше і більше залежать від нейромереж, оскільки ці технології виявляються незамінними у багатьох сферах життя. Вони використовуються для покращення штучного інтелекту, автоматизації процесів, вдосконалення медичної діагностики, розробки нових технологій у сфері транспорту, торгівлі та багатьох інших галузях. Нейромережі допомагають зробити рішення швидшими, точнішими та більш ефективними, що стає ключовим фактором у сучасному світі, де швидкість та інновації відіграють важливу роль.

Проведено аналіз згортальних нейронних мереж і методів, що використовуються в них. Далі, увага була зосереджена на використанні згортальних нейронних мереж для забезпечення безпеки при розпізнаванні об'єктів на фотографіях та відеопотоці. І наостанок, я дослідив порівняльний аналіз алгоритмів методів згортальних нейронних мереж, з метою, визначити їх ефективність та можливі області застосування.

Метою роботи є розробка ефективної системи розпізнавання зображень на базі методів, заснованих на згортальній нейромережі.

Об'єктом дослідження є процес розпізнавання зображень

Предметом дослідження є методи розпізнавання зображень на основі згортальній нейромережі.

Завданням дослідження було:

- дослідити та аналізувати згортальні нейромережі та їх методи;
- використати згортальні нейромережі, для безпеки розпізнавання об'єктів у фото та відео потоці;
- зрівняти алгоритми методів згортальних нейромереж.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 3 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1 СУЧАСНІ СИСТЕМИ ЗГОРТАЛЬНИХ НЕЙРОМЕРЕЖ

1.1 Опис сучасних згортальних нейромереж

EfficientNet — це архітектура згорткової нейронної мережі та метод масштабування, який рівномірно масштабує всі розміри глибини/ширини/роздільної здатності за допомогою складеного коефіцієнта. На відміну від звичайної практики, яка довільно масштабує ці фактори, метод масштабування EfficientNet рівномірно масштабує ширину, глибину та роздільну здатність мережі за допомогою набору фіксованих коефіцієнтів масштабування. Наприклад, якщо ми хочемо використовувати 2^N разів більше обчислювальних ресурсів, то ми можемо просто збільшити глибину мережі на α^N , ширину на β^N , а розмір зображення на γ^N , де $\alpha, \beta, \gamma, \epsilon$ постійними коефіцієнтами, визначеними пошуком на дрібній сітці на вихідній дрібній моделі. EfficientNet використовує складений коефіцієнт ϕ для рівномірного принципового масштабування ширини, глибини та роздільної здатності мережі.

EfficientDet — це тип моделі виявлення об'єктів, яка використовує оптимізаційні і магістральні налаштування. Загальна архітектура EfficientDet у значній мірі відповідає парадигмі одноступінчастих (одноступеневих) детекторів. На основі взятого EfficientNet, попереднього на ImageNet, до якого приєднаний шар зі зведеною двонаправленою пірамідою визнань (BiFPN), за якою йде класова і блочна мережа для генерації передбаченого класу об'єкта і обмежувального прямокутника відповідно.

EfficientDet – one-stage детектор, використовує 4 основні мережі:

- Backbone network;
- Feature network;
- Bounding Box prediction network;
- Class prediction network.

Backbone network – мережа для отримання ознак. Як backbone

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 4 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

використовується EfficientNet, він:

- забезпечує кращу точність;
- підвищує ефективність (зменшення FLOP);
- зменшено кількість параметрів.

SpineNet — це магістраль згорткової нейронної мережі з проміжними функціями, зміненими масштабом, і міжмасштабними зв'язками, які вивчаються під час завдання виявлення об'єктів за допомогою Neural Architecture Search. SpineNet – це магістральна нейросетєва архітектура для завдань розпізнавання об'єкта. Модель створює найсучасніші підходи до задачі розпізнавання об'єктів на даних COCO.

При цьому SpineNet виробляє на 60% менше збільшення і отримує ResNet-FPN на 6% від AP. Запропоновану архітектуру також використовують для класифікації. SpineNet створює найсучасніші дані на 6% у точності в даній iNaturalist.

CenterTrack — можна описати так: він має bounding boxes і з цих боксів він може отримати центр нашого об'єкта. Отримавши цей центр, він може вирішити не завдання detection з проорокуванням чотирьох координат і класу, як завжди, а просто знаходити такі точки, і з них потім регресувати наші bounding boxes.

ThunderNet — це двоетапна модель виявлення об'єктів. Проект ThunderNet спрямований на обчислювально дорогі структури в найсучасніших двоступеневих детекторах. Магістраль використовує мережу SNet на основі ShuffleNetV2, призначену для виявлення об'єктів. У частині виявлення ThunderNet дотримується конструкції головки виявлення в Light-Head R-CNN і додатково стискає підмережі RPN і R-CNN.

Щоб усунути зниження продуктивності, спричинене невеликими магістралями та невеликими картами функцій, ThunderNet використовує два нові ефективні блоки архітектури: модуль покращення контексту (CEM) і модуль просторової уваги (SAM). CEM поєднує карти функцій із кількох масштабів, щоб використовувати інформацію

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 5 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

про локальний і глобальний контекст, тоді як SAM використовує інформацію, отриману в RPN, щоб уточнити розподіл функцій у викривленні RoI.

DenseNet — це щільна згорточна мережа, яка з'єднує кожен шар з кожним іншим у прямому зв'язку. Тоді як традиційні згорточні мережі з L шарами мають L з'єднань – по одному між кожним шаром і його наступним шаром – наша мережа має $L(L+1)/2$ прямих з'єднань. Для кожного шару карти функцій усіх попередніх шарів використовуються як вхідні дані, а власні карти функцій використовуються як вхідні дані для всіх наступних шарів.

DenseNets мають кілька переконливих переваг: вони усувають проблему зникаючого градієнта, підсилюють розповсюдження ознак, заохочують повторне використання функцій і суттєво зменшують кількість параметрів.

CSPDenseNet — основна мета проектування CSPNet полягає в тому, щоб ця архітектура могла досягти багатшої комбінації градієнтів, одночасно зменшуючи кількість обчислень. Ця мета є досягається шляхом поділу карти функцій базового шару на дві частини, а потім об'єднати їх через запропоновану міжступеневу ієрархію. Наша головна концепція — створити градієнтний потік, що поширюється різними мережевими шляхами шляхом розщеплення градієнтного потоку.

Поширена градієнтна інформація може мати велику різницю кореляції через перемикання конкатенації та переходу кроки. Крім того, CSPNet може значно зменшити суму обчислень, а також покращити швидкість логічного висновку

DetNASNet — розроблений спеціально для виявлення об'єктів, виявлених за допомогою пошуку архітектури DetNAS. Він використовує блоки ShuffleNet V2 як основний будівельний блок, що дозволяє йому бути більш ефективним в програмах реального часу, також DetNASNet забезпечує високу точність із середньою mAP - (mean Average Precision) 42,3% на наборі даних COCO, перевершуючи інші найсучасніші методи виявлення об'єктів. Крім того, використання FPN дозволяє краще виявляти об'єкти в різних масштабах на зображенні, підвищуючи його високу загальну точність.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 6 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1.2 Методи роботи сучасних згортальних нейромереж

EfficientNet — використовувала метод автоматично за допомогою AutoML фреймворку - MNAS. MNAS під час вибору пристрою нейромережі оптимізує і точність моделі, і її розмір одночасно (FLOPS). У результаті початкова модель використовувала архітектуру MBConv, схожу на MobileNetV2 і MnasNet.

Початкова модель потім масштабувалася і породила клас EfficientNets моделей. складного масштабування, користується він інтуїцією, що якщо вхідне зображення більше, то мережі потрібно більше шарів, щоб збільшити сприйнятливий поле, і більше каналів, щоб захопити більш дрібнозернисті візерунки на більшому зображенні.

Базова мережа EfficientNet-V0 базується на перевернутих блоках залишкових вузьких місць MobileNetV2, на додаток до блоків стискання та збудження.

EfficientNets також добре передавала та досягала найсучаснішої точності на CIFAR-100 (91,7%), Flowers (98,8%) та 3 інших наборах даних для навчання передачі з на порядок меншою кількістю параметрів.

На рисунку 1.1 представлено масштабування моделі.

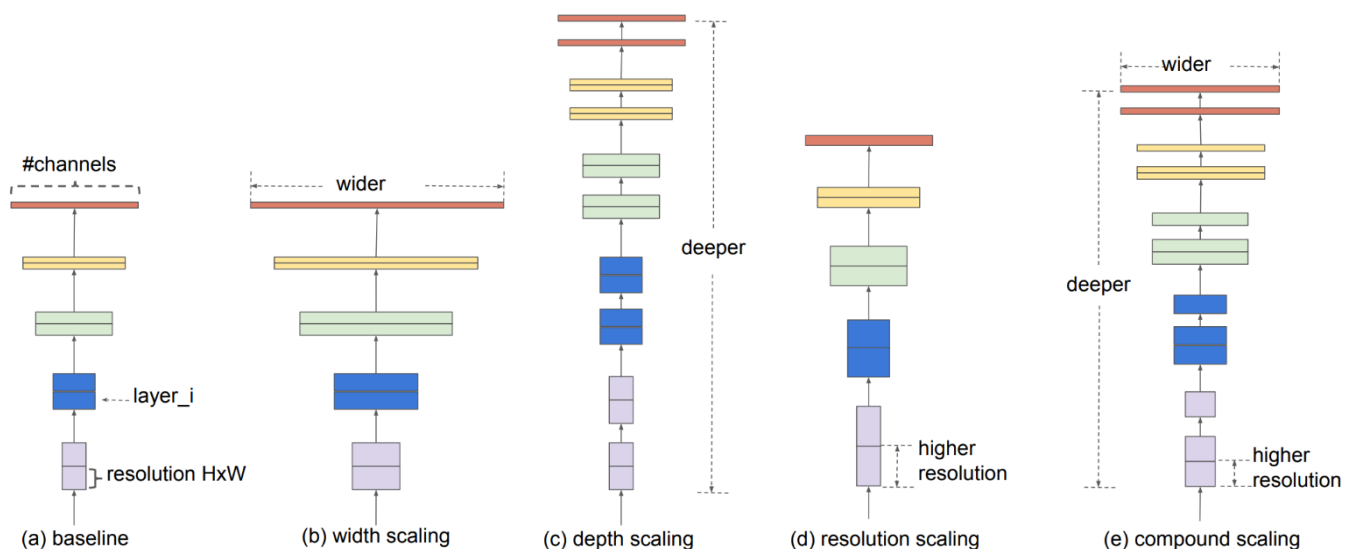


Рисунок 1.1 – Масштабування моделі

(а) є прикладом базової мережі;

(b)-(d) є звичайним масштабуванням, яке збільшує лише один вимір мережі, ширину, глибину або роздільну здатність.

(е) є запропонований нами комплексний метод масштабування, який рівномірно масштабує всі три виміри з фіксованим співвідношенням.

На рисунку 1.2 представлено масштабування моделі.

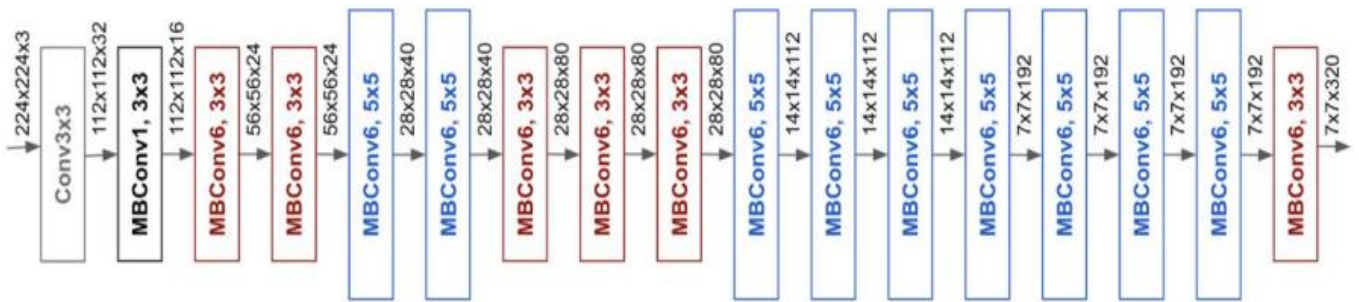


Рисунок 1.2 – Архітектура початкової моделі EfficientNet-B0

EfficientDet використовує метод BiFPN із складного методу масштабування, який рівномірно масштабує роздільну здатність, глибину та ширину для всіх магістралей, мереж функцій і передбачення коробок/класів мережі одночасно.

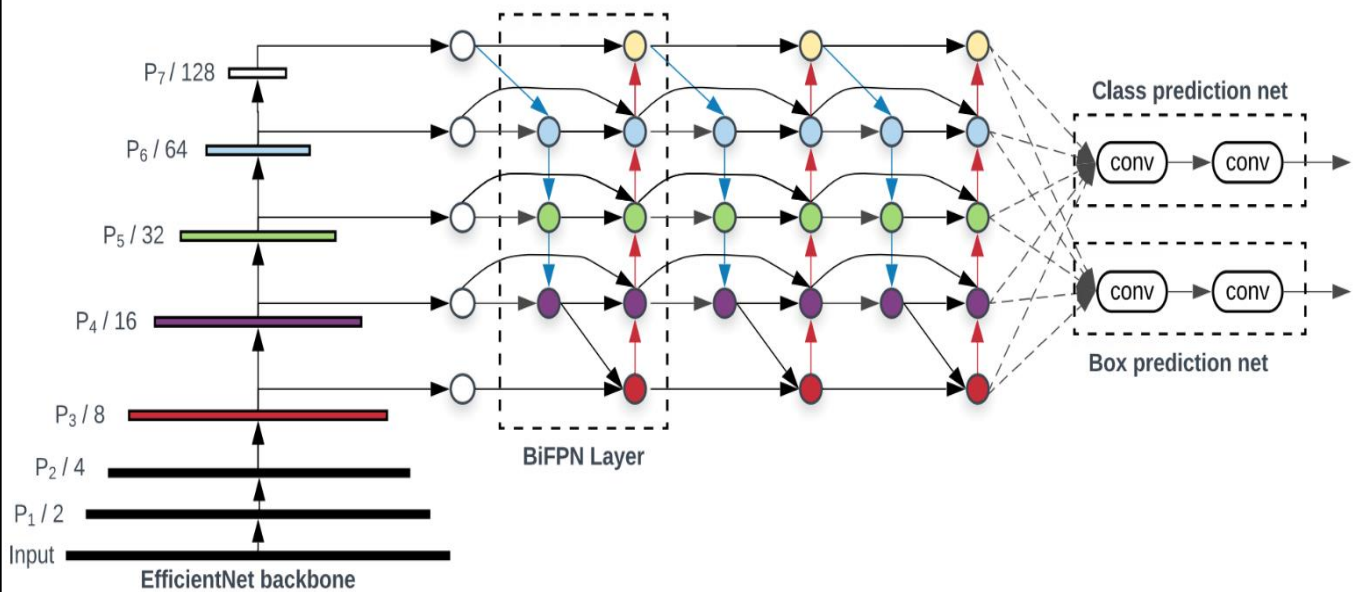


Рисунок 1.3 – Архітектура EfficientDet

У ній використовується EfficientNet як магістральна мережа, ViFPN як мережа функцій і спільна мережа прогнозування класів/блоків.

Як рівні ViFPN, так і рівні мережі класів/блоків повторюються кілька разів на основі різних обмежень ресурсів.

SpineNet використовує метод NAS (Neural Architecture Search) у ній використовується контролер навчання із підкріпленням. Він пропонує різні архітектури, що вирушають у середу, де повноцінно навчаються.

Точність висновку грає роль винагороди, і рішення про вибір архітектури залежатиме саме від неї.

Архітектура SpineNet містить мережу з зменшенням масштабу, що супроводжується навченою мережею з масштабом, що переставляється.

Область пошуку NAS для побудови мережі з масштабом, що переставляється, включає в себе переставлення масштабів, між масштабних зв'язків і підстроювання блоків.

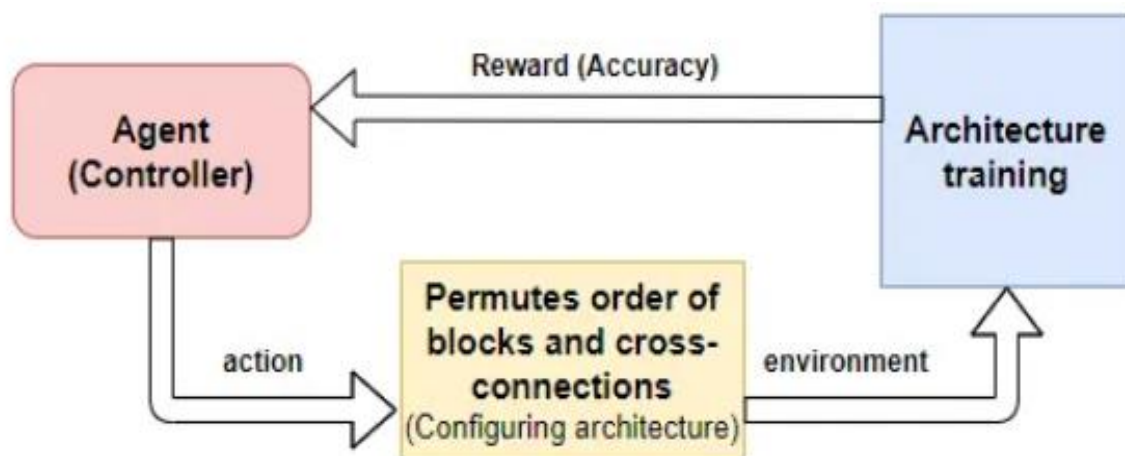


Рисунок 1.4 – Метод пошуку нейронної архітектури

Також SpineNet використовувала метод ResNet модель з масштабами, що переставляється, формується за допомогою перестановки блоків архітектури ResNet. Для порівняння мережі з масштабом, що зменшується, з мережею, що

працює за принципом перестановки, генерується ряд проміжних моделей, які покроково зміщують архітектуру мережі в бік її перестановки форми.

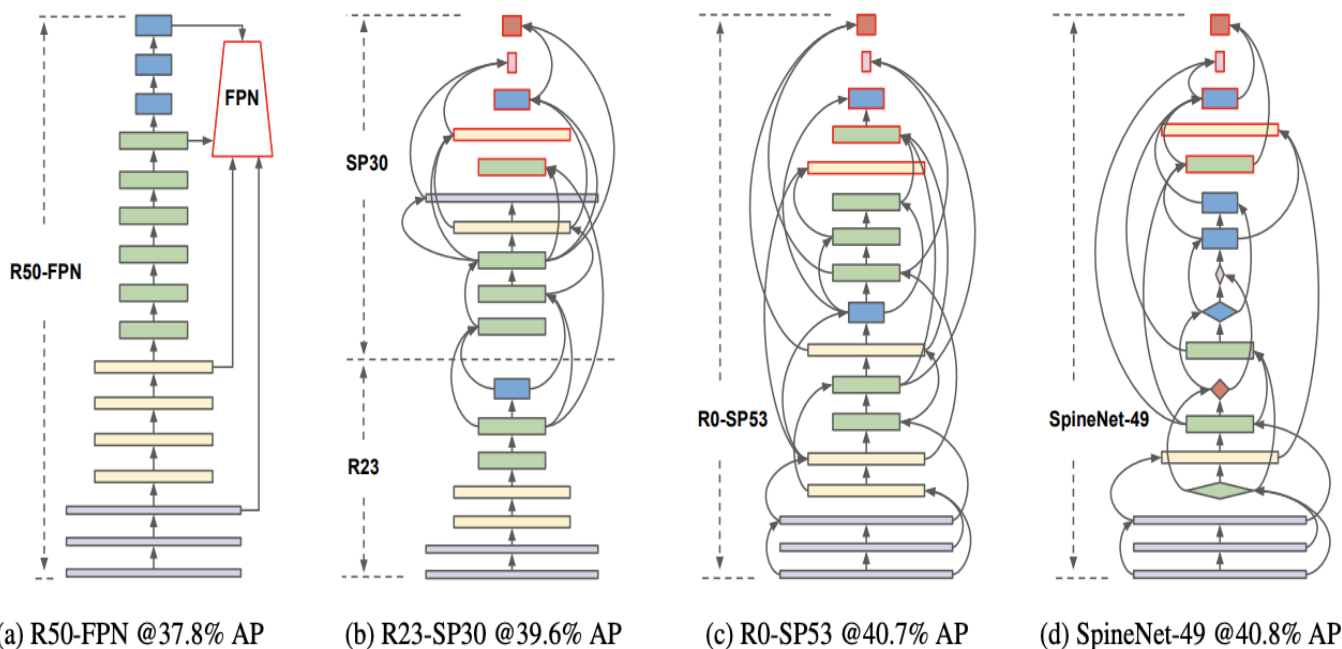


Рисунок 1.5 – Побудова мережі з перестановкою масштабу шляхом перестановки ResNet

Від (а) до (d) обчислення поступово переходять від ResNet-FPN до масштабно переставлених мереж.

(а) Модель R50-FPN, яка витрачає більшість обчислень у ResNet-50, а потім FPN, досягає 37,8% AP;

(b) R23-SP30, інвестуючи 7 блоків у ResNet і 10 блоків у мережу з перестановкою масштабу, досягає 39,6% AP;

(c) RO-SP53, інвестуючи всі блоки в мережу з перестановкою масштабу, досягає 40,7% AP;

(d) Архітектура SpineNet-49 досягає 40,8% AP з 10% менше FLOPS (85,4B проти 95,2B) шляхом вивчення додаткових коригувань блоків. Прямокутний блок представляє вузьке місце, а ромбовидний блок – залишковий блок. Вихідні блоки позначені червоною рамкою.

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

У частині виявлення RPN стискається, а підмережа R-CNN використовує 1024-d шар fc для кращої ефективності. Модуль покращення контексту використовує семантичну та контекстну інформацію з різних масштабів. Модуль просторової уваги вводить інформацію з RPN для уточнення розподілу ознак.

DenseNet використовує метод на базі CNN (Convolutional Neural Network) але він бере вихідну карту функцій і об'єднує її з усіма попередніми вхідними картами функцій, як на рисунку 1.8. Це максимізує потік інформації між шарами, зменшує загальну кількість параметрів, а також значно покращує будь-які зникаючі градієнти.

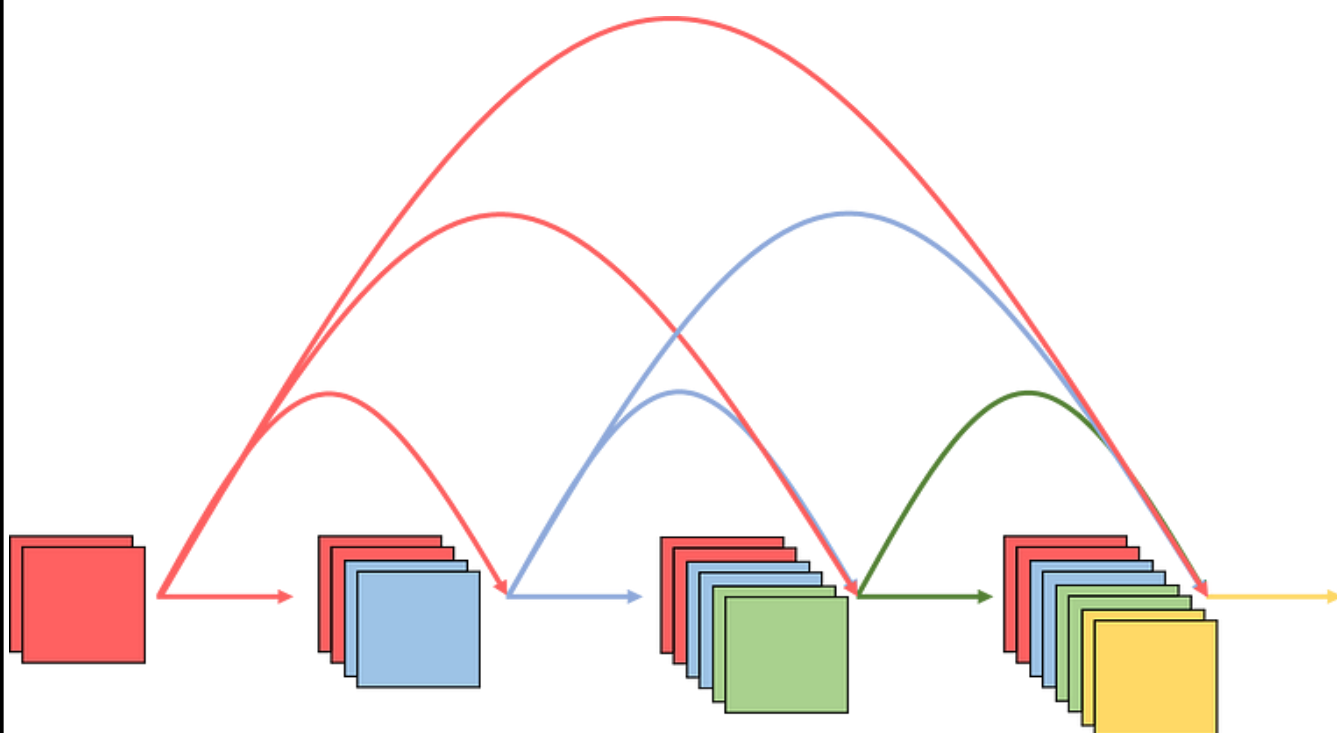
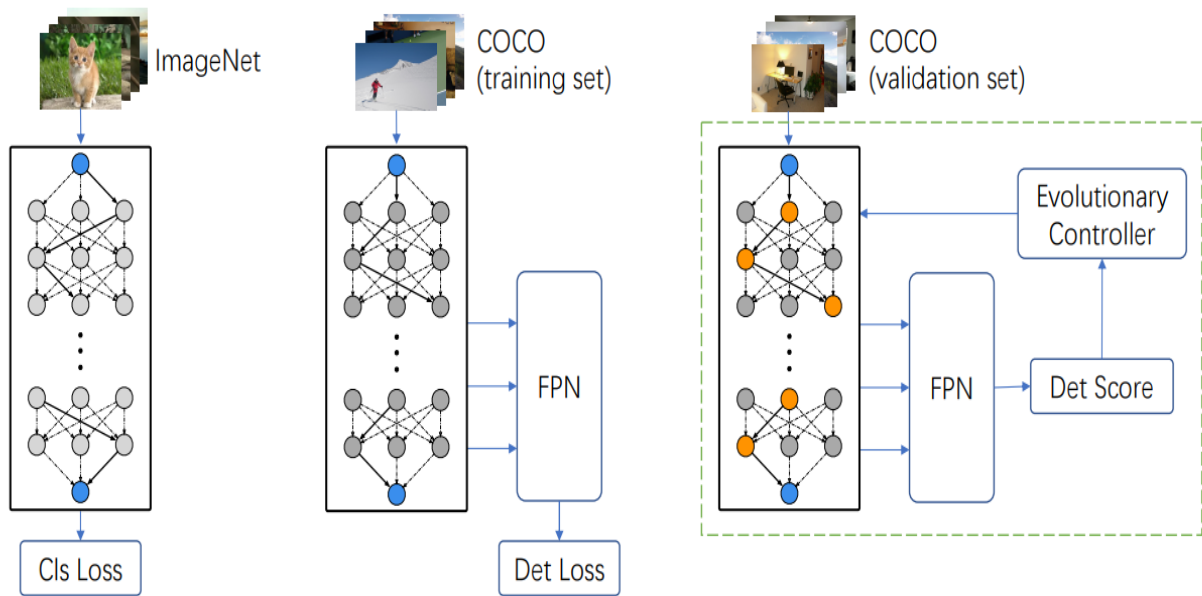


Рисунок 1.8 – DenseNet зі швидкістю зростання

DenseNet це складена функція, тому вона складалася з трьох послідовних операцій: пакетної нормалізації BN - (Batch Normalization), за якою слідувала випрямлена лінійна одиниця ReLU - (Rectified Linear Unit), а потім згортка 3x3 . Це і було те, що ми називаємо щільним блоком . Їх схематично можна побачити на рисунку 1.9.

детекторах об'єктів неможливо. Типовий графік навчання детектора вимагає магістральні мережі, які мають бути попередньо навчені за допомогою ImageNet.



Step1: Supernet pre-training Step2: Supernet fine-tuning Step3: Evolutionary search on the trained supernet

Рисунок 1.12 – Конвеєр DetNAS, який шукає магістралі в детекторах об'єктів

Існує три кроки: попереднє навчання супермережі на ImageNet, тонке налаштування супермережі на наборі для навчання виявлення, наприклад, COCO, і пошук архітектури в навченій супермережі за допомогою алгоритму еволюції. Набір перевірки фактично відокремлений від COCO і складається з 5k зображень.

1.3 Порівняння згортальних нейромереж

EfficientNet - моделі значно перевищують на рисунку 1.13 конкуруючі архітектури. EfficientNet-B7 обіграла розглянуті згорткові архітектури та стала топ-1 за точністю для завдання. EfficientNet-B1 в 7.6x менше та в 5.7x швидше.

EfficientDet - EfficeintDet-D0 досягає 33,8 AP на наборі даних COCO з 2,5B FLOPS на рисунку 1.14.

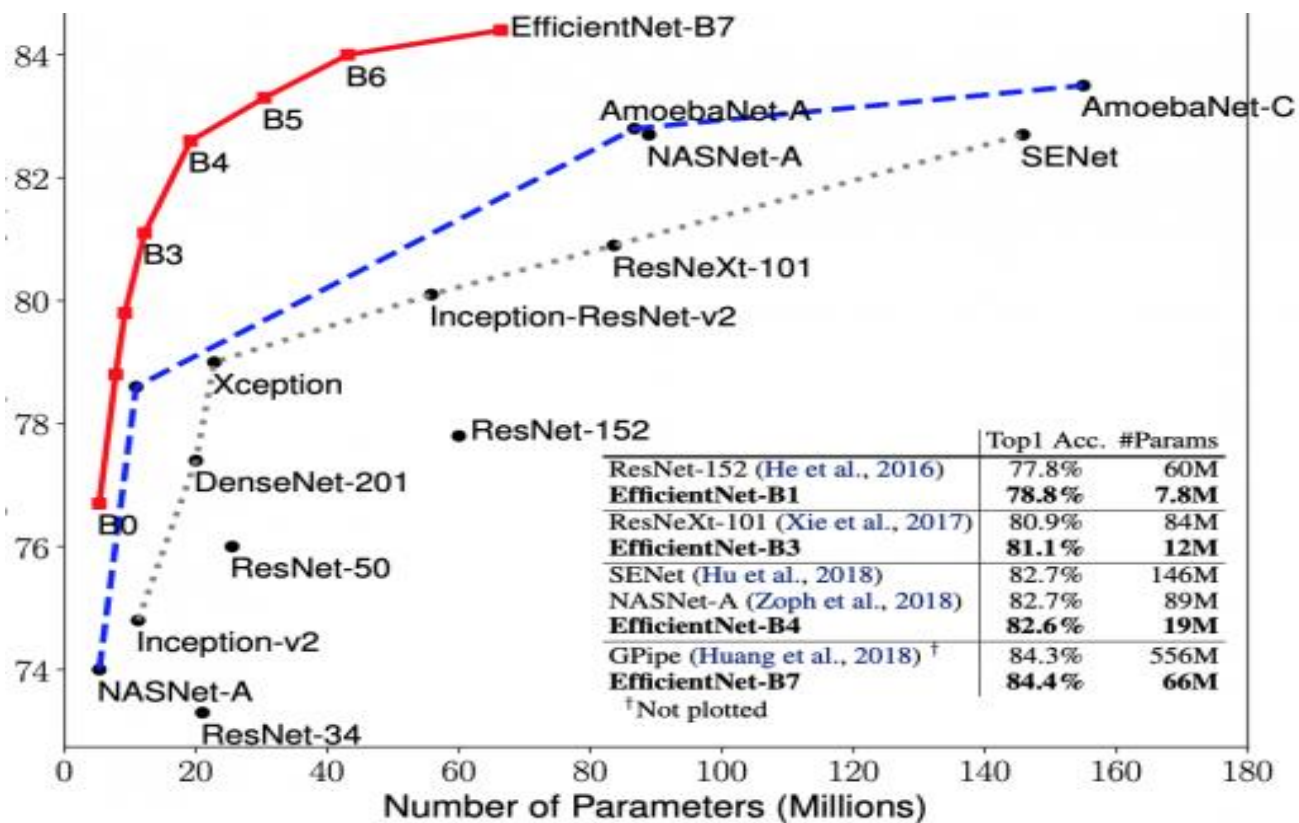


Рисунок 1.13 – Порівняння EfficientNet

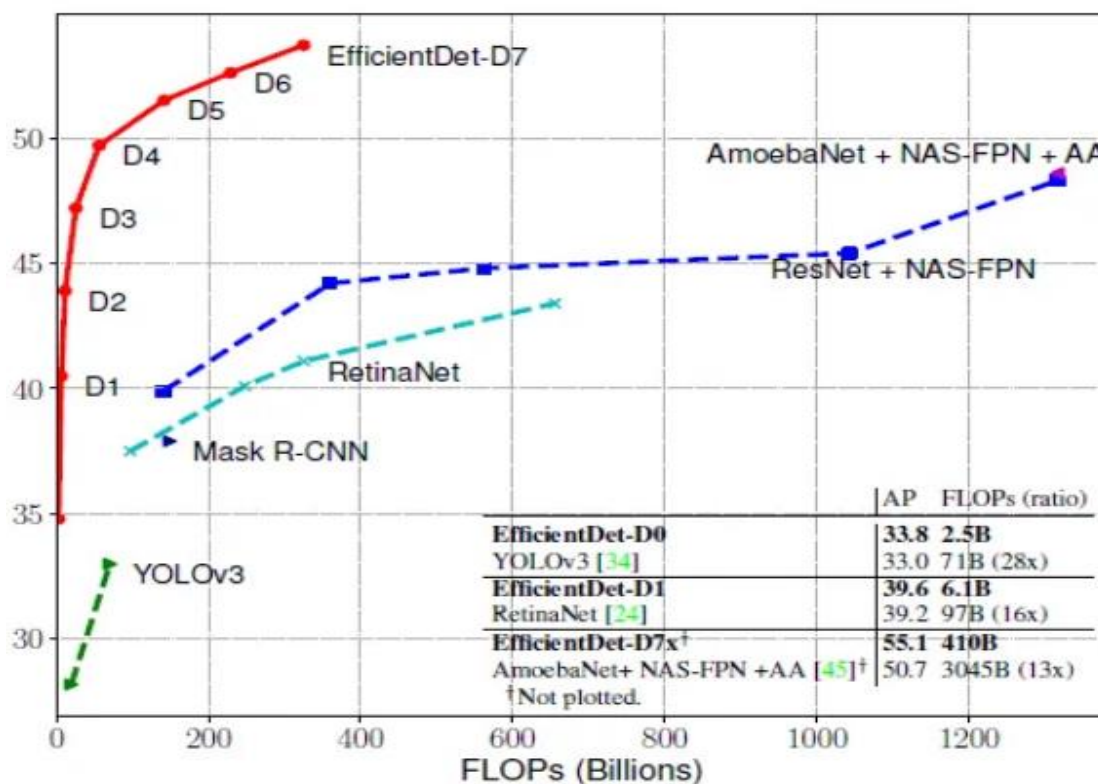


Рисунок 1.14 – Порівняння EfficientDet

SpineNet - досягає більш високого показника AP на рисунку 1.15 при значно меншій кількості операцій з плаваючою точкою в порівнянні з ResNet-FPN і NAS_FPN.

Результати в наступній таблиці отримані на COCO val2017 і показують, що SpineNet-49 вимагає приблизно на 10% менше операцій на рисунку 1.16.3 плаваючою точкою, а показник AP покращився до 40,8% порівняно з 37,8% R50-FPN.

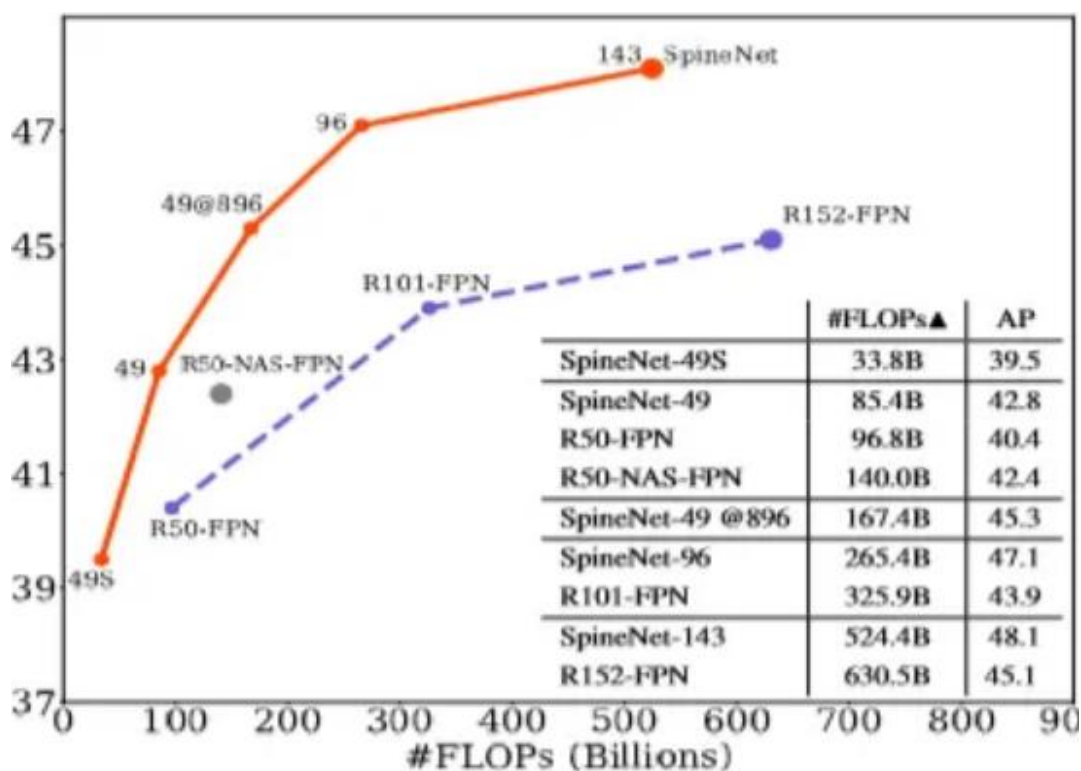


Рисунок 1.15 – Порівняння SpineNet

| model | block adju. | #FLOPs | AP |
|-------------|-------------|--------|------|
| R50-FPN | - | 96.8B | 37.8 |
| R35-SP18 | - | 91.7B | 38.7 |
| R23-SP30 | - | 96.5B | 39.7 |
| R14-SP39 | - | 99.7B | 39.6 |
| R0-SP53 | - | 95.2B | 40.7 |
| SpineNet-49 | ✓ | 85.4B | 40.8 |

Рисунок 1.16 – Порівняння SpineNet

CenterTrack в порівнянні з іншими моделями руху на дата сетах MOT17, KITTI та nuScenes. Нижче на рисунку 1.17 видно, що видає порівнянні чи вищі результати, ніж альтернативні підходи.

| | MOT17 | | | | KITTI | | | | nuScenes | |
|---------------|-------------|------|-------|-------|-------------|------|------|-------------|-------------|------------|
| | MOTA↑ | FP↓ | FN↓ | IDSW↓ | MOTA↑ | FP↓ | FN↓ | IDSW↓ | AMOTA@0.2↑ | AMOTA@1↑ |
| no motion | 65.8 | 4.5% | 28.4% | 1.3% | 87.1 | 5.4% | 5.8% | 1.6% | 17.8 | 3.6 |
| Kalman filter | 66.1 | 4.5% | 28.4% | 1.0% | 87.9 | 5.4% | 5.8% | 0.9% | 18.3 | 3.8 |
| optical flow | 66.1 | 4.5% | 28.4% | 1.0% | 88.4 | 5.4% | 5.8% | 0.4% | 26.6 | 6.2 |
| ours | 66.1 | 4.5% | 28.4% | 1.0% | 88.7 | 5.4% | 5.8% | 0.1% | 28.3 | 6.8 |

Рисунок 1.17 – Порівняння CenterTrack

ThunderNet Порівняння і попереднього полегшеного детектори на COCO test-dev. ThunderNet досягає покращень як точністю, так і ефективністю на рисунку 1.18.

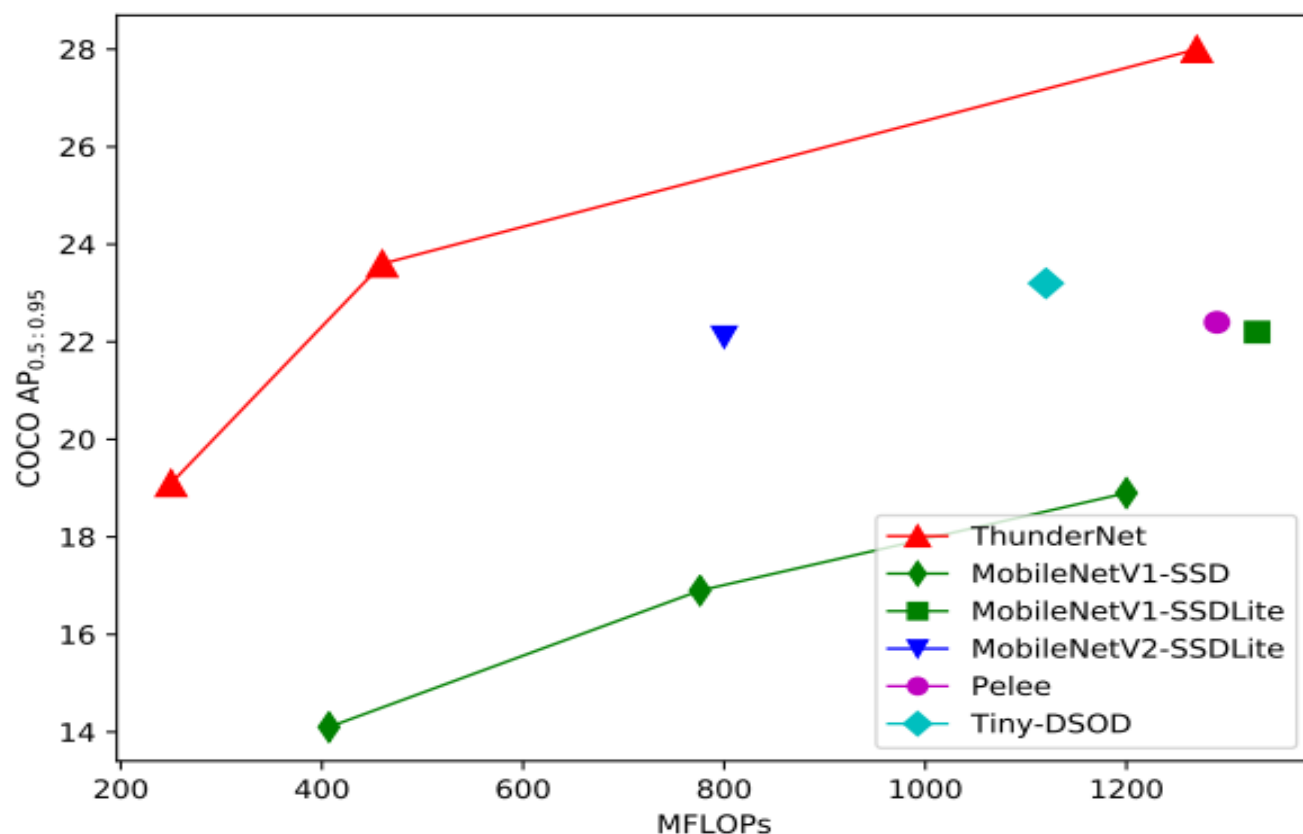


Рисунок 1.18 – Порівняння ThunderNet

CSPDenseNet особливо потужні, коли вони стають глибшими [8, 37, 11] і ширшими [38] на рисунку 1.19.

DenseNet - навіть із зменшеною кількістю параметрів модель на рисунку 1.20. має значно нижчу помилку перевірки для моделі ResNet із такою ж кількістю параметрів. Ці експерименти проводилися на обох моделях із гіперпараметрами, які більше підходили для ResNet.

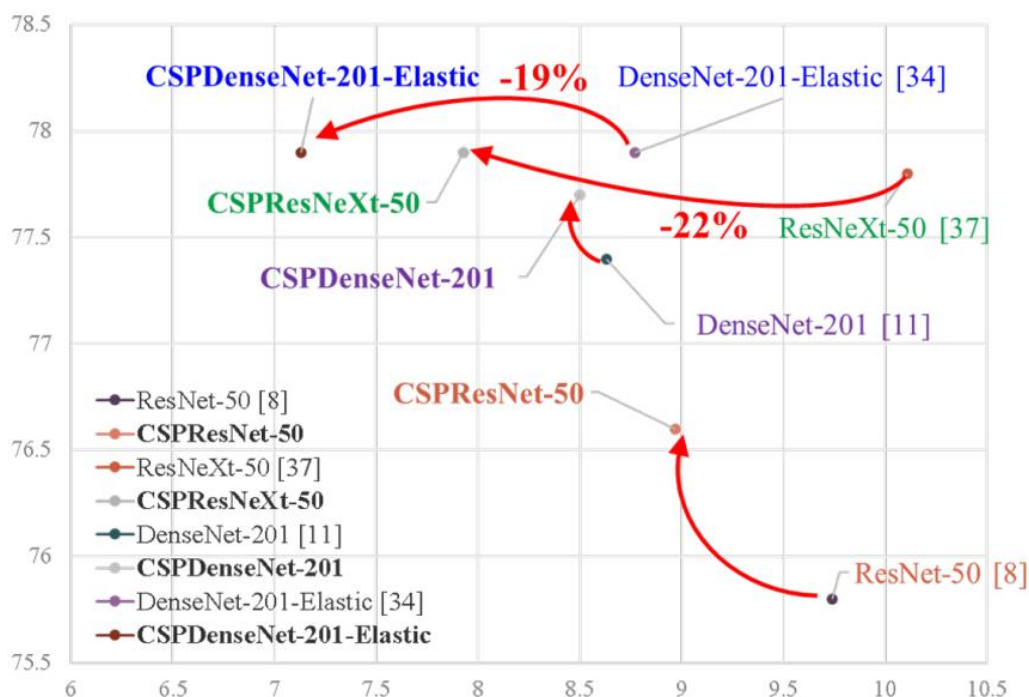


Рисунок 1.19 – Порівняння CSPDenseNet

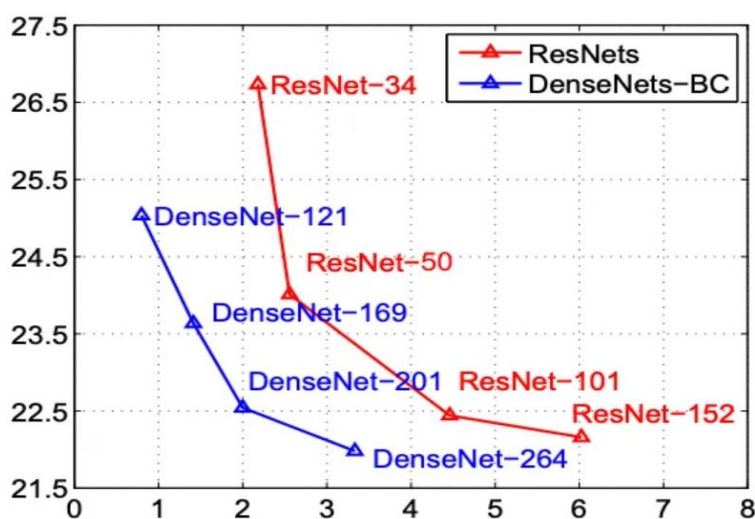


Рисунок 1.20 – Порівняння DenseNet

1.4 Висновки

У першому розділі було детально розглянуто сучасні системи згортальних нейромереж, як вони автоматично використовують фреймворки, беруть за основу другі методи, оптимізуючи їх та використовуючи їх краще. Також було проведено аналіз та опис їх основних характеристик, методів роботи, та проведено порівняння різних підходів до роботи з неординаковими кінцевим результатом.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 21 |

2 ДОСЛІДЖЕННЯ МЕТОДІВ ТА АНАЛІЗУ ЗГОРТАЛЬНИХ НЕЙРОМЕРЕЖ

2.1 Початок роботи згортальних нейромереж

Хоча суперкомп'ютер IBM Deep Blue переміг чемпіона світу з шахів Гаррі Каспарова ще в 1996 році, лише зовсім недавно комп'ютери змогли надійно виконувати, здавалося б, тривіальні завдання, такі як виявлення цуценя на зображенні або розпізнавання вимовлених слів. Чому ці завдання такі легкі для нас, людей? Відповідь полягає в тому, що сприйняття в основному відбувається поза сферою нашої свідомості, в межах спеціалізованих зорових, слухових та інших сенсорних модулів нашого мозку. До того часу, коли сенсорна інформація досягає нашої свідомості, вона вже прикрашена характеристиками високого рівня; наприклад, коли ви дивитеся на фотографію милого цуценя, ви не можете не бачити цуценя, не помічати його миловидності. Ви також не можете пояснити, як ви впізнаєте милого цуценя; це просто для вас очевидно. Таким чином, ми не можемо довіряти нашому суб'єктивному досвіду: сприйняття зовсім не тривіальне, і щоб зрозуміти його, ми повинні подивитися, як працюють сенсорні модулі.

Згорткові нейронні мережі (CNN) виникли в результаті дослідження зорової кори головного мозку, і вони використовуються для розпізнавання зображень з 1980-х років. В останні кілька років, завдяки збільшенню обчислювальної потужності, обсяг доступного навчання даних і трюки для навчання глибоких мереж CNN, вдалося досягти надлюдської продуктивності в деяких складних візуальних завданнях. Вони забезпечують служби пошуку зображень, безпілотні автомобілі, автоматичні системи класифікації відео тощо. Крім того, CNN не обмежуються візуальним сприйняттям: вони також успішні для багатьох інших завдань, таких як розпізнавання голосу та обробка природної мови. Архітектура зорової кори головного мозку Девід Х. Хьюбель і Торстен Візель провели серію експериментів на котах у 1958 та 1959 роках (а через кілька років на мавпах), що

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 22 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

дало важливе розуміння структури зорової кори (автори отримали Нобелівську премію).

Премія з фізіології та медицини в 1981 році за їхню роботу). Зокрема, вони показали, що багато нейронів у зоровій корі мають невелике локальне рецептивне поле, тобто вони реагують лише на зорові стимули, розташовані в обмеженій області поля зору на рисунку 2.1, на якому локальні рецептивні поля п'яти нейронів представлені пунктирними колами). Рецептивні поля різних нейронів можуть перекриватися, і разом вони утворюють все поле зору. Крім того, автори показали, що деякі нейрони реагують лише на зображення горизонтальних ліній, тоді як інші реагують лише на лінії з різною орієнтацією (два нейрони можуть мати однакове рецептивне поле, але реагувати на різні орієнтації ліній). Вони також помітили, що деякі нейрони мають більші рецептивні поля, і вони реагують на більш складні патерни, які є комбінаціями патернів нижчого рівня. Ці спостереження привели до ідеї, що нейрони вищого рівня базуються на результатах сусідніх нейронів нижчого рівня (зверніть увагу, що на рисунку 2.1 кожен нейрон підключений лише до кількох нейронів з попереднього шару). Ця потужна архітектура здатна виявляти всілякі складні візерунки в будь-якій області поля зору.

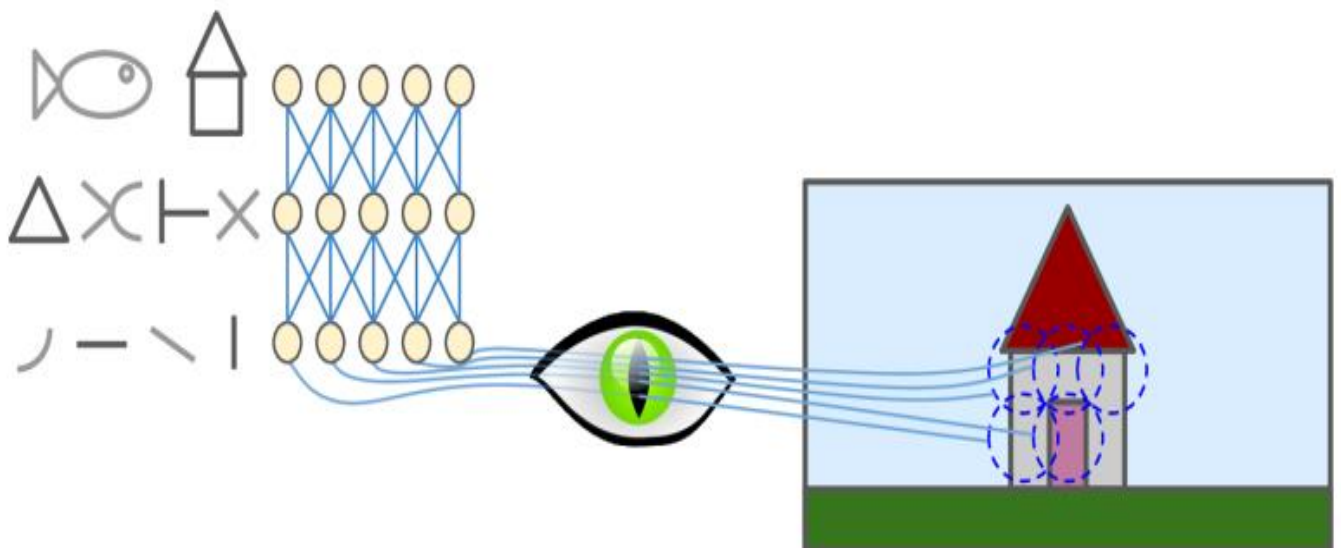


Рисунок 2.1 – Узагальнений спосіб дії нейронів зорової кори

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

для кожного вхідного пікселя зображення. Візьмемо, наприклад, кольорове зображення розміром (224x224x3) на рисунку 2.4. Вхідний шар матиме 150528 нейронів. Якщо ми матимемо лише три приховані шари невеликого розміру зі 128 нейронами кожен, ми перевищимо 300 мільярдів параметрів для навчання. CNN вирішують цю проблему за допомогою частково пов'язаних шарів і розподілу ваги.

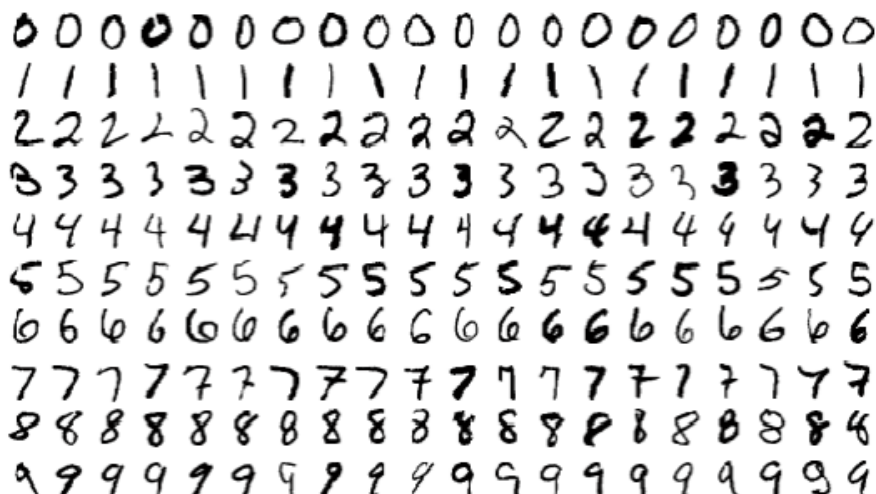


Рисунок 2.3 – Візуалізація невеликої підмножини MNIST - навчального набору даних

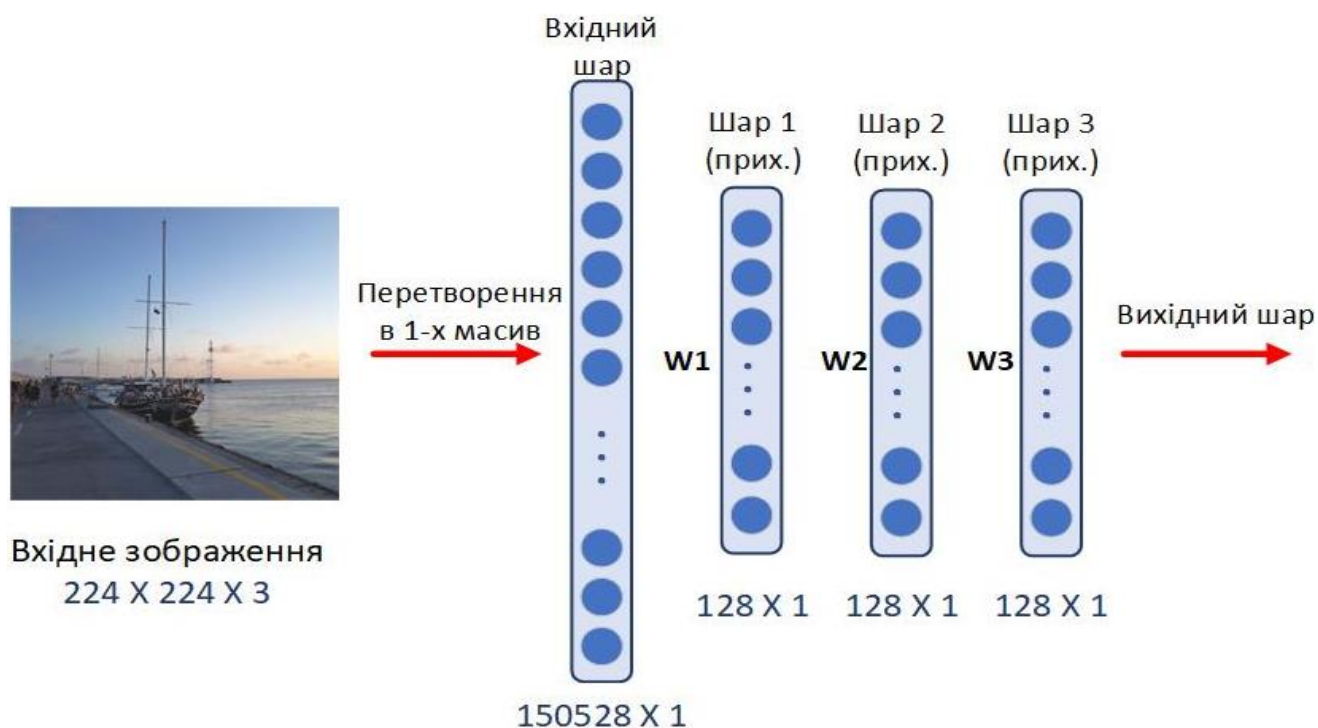


Рисунок 2.4 – Нейронна обробка зображення

2.2 Згорткові шари та їх робота

Згорткові шари

Найважливішим будівельним блоком CNN є згортковий рівень: нейрони в першому згортковому шарі підключені не до кожного окремого пікселя вхідного зображення (як це було в шарах, розглянутих у попередніх розділах), а лише до пікселів у їх сприйнятливому полі на рисунку 2.5. У свою чергу, кожен нейрон другого згорткового шару пов'язаний лише з нейронами, розташованими в маленькому прямокутнику першого шару. Ця архітектура дозволяє мережі зосередитися на невеликих елементах низького рівня в першому прихованому шарі, а потім зібрати їх у більші функції вищого рівня в наступному прихованому шарі і так далі. Ця ієрархічна структура поширена в зображеннях реального світу, що є однією з причин, чому CNN так добре працюють для розпізнавання зображень.

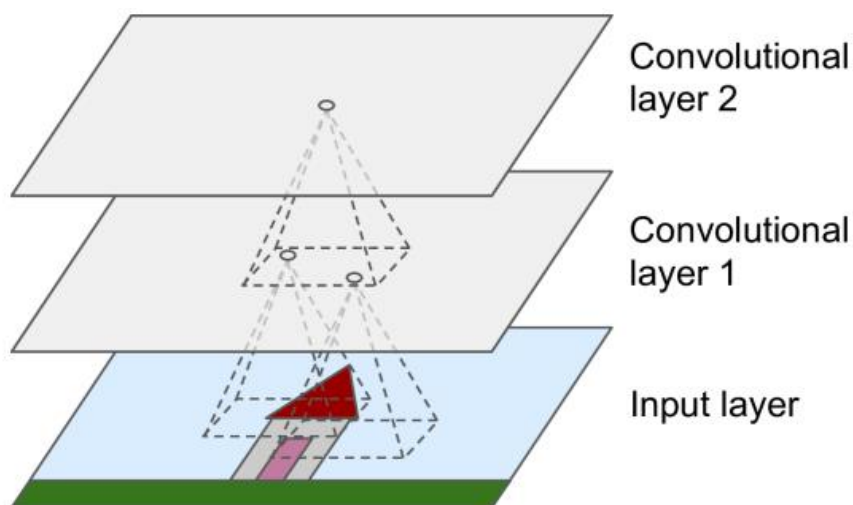


Рисунок 2.5 – Шари CNN з прямокутними локальними рецептивними полями

Це концепція згорткової нейронної мережі: зменшення розмірів даних для вилучення найбільш відповідних функцій. Усі багатошарові нейронні мережі, які розглядалися досі, мали шари, що склалися з довгої лінії нейронів, перш ніж подавати їх у нейронну мережу. довелося звести вхідні зображення до 1D. У CNN

кожен шар представлений у 2D, що полегшує зіставлення нейронів із відповідними входами.

Нейрон, розташований у рядку i , стовпці j даного шару, з'єднаний з виходами нейронів попереднього шару, розташованих у рядках $i - i + f_h - 1$, стовпцях $j - j + f_w - 1$, де f_h і f_w є висота і ширина рецептивного поля на рисунку 2.6. Крім того, у згортковій нейронній мережі не всі нейрони з'єднані, як у щільних шарах. Згортковий шар моделюється на основі того, як зорова кора працює в нашому мозку та як наш мозок обробляє те, що бачать наші очі.

Наші клітини працюють так, що групи клітин з обмеженим полем зору реагують на світло, яке потрапляє на задню частину наших очей, і всі ці групи клітин з обмеженим полем зору пов'язані з нейронами в мозку. де вони обробляються та складаються в одне велике зображення, яке ми розуміємо. За допомогою згорткової нейронної мережі ми маємо один нейрон на піксель зображення. Ми також визначаємо розмір ядра шару, який означає, скільки нейронів ми хочемо щоб було в цих «групах», які складають візуальні поля, та зображення. Виходи однієї групи нейронів, які складають ядро, підключаються до одного входу на попередньому рівні. Таким чином, зображення обробляється, коли згортковий шар згортає (тобто переміщує) зображення, обробляючи кожену групу пікселів і передаючи вихідні дані попередньому шару для обробки. Це і називається нульовим заповненням.

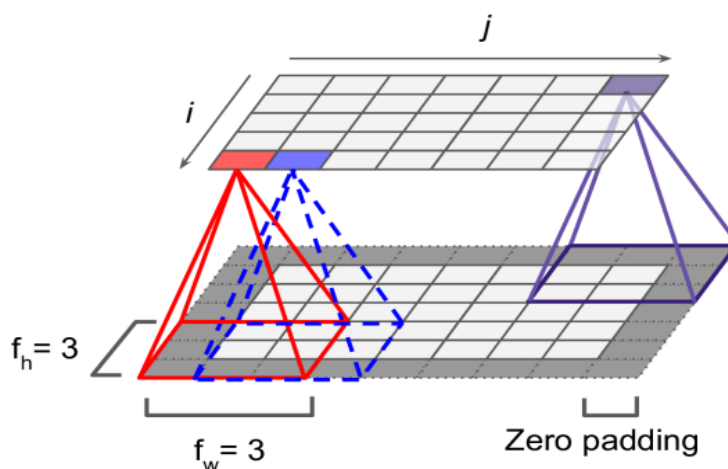


Рисунок 2.6 – Зв'язки між шарами та нульове заповнення

Також можна з'єднати великий вхідний рівень із набагато меншим шаром, розділивши сприйнятливі поля, як показано на рисунку 2.7. Це значно зменшує обчислювальну складність моделі. Перехід від одного рецептивного поля до іншого називається кроком. На діаграмі вхідний шар 5×7 (плюс нульове заповнення) з'єднано з шаром 3×4 , використовуючи сприйнятливі поля 3×3 і крок 2 (у цьому прикладі крок однаковий в обох напрямках, але не повинно бути так). Нейрон, розташований у рядку i , стовпці j у верхньому шарі, з'єднаний з виходами нейронів попереднього шару, розташованих у рядках $i \times s_h$ до $i \times s_h + f_h - 1$, стовпцях $j \times s_w$ до $j \times s_w + f_w - 1$, де s_h і s_w – вертикальний і горизонтальний кроки.

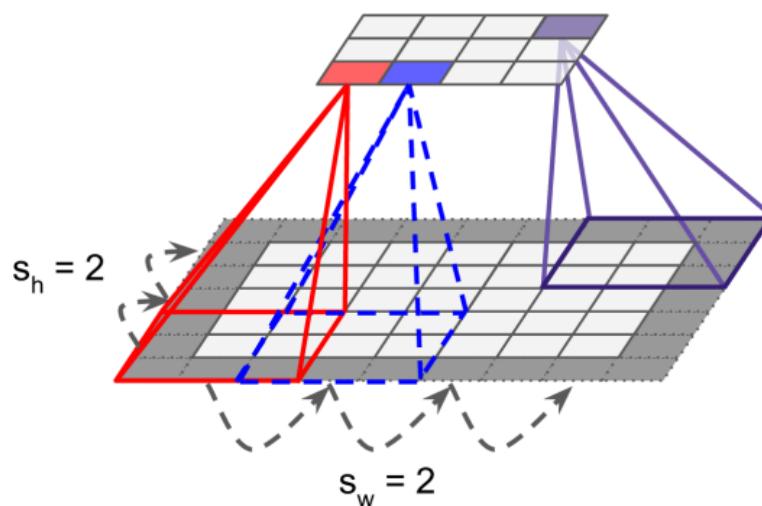


Рисунок 2.7 – Зменшення розмірності

Фільтри

Ваги нейрона можна представити у вигляді маленького зображення розміром з рецептивне поле. Наприклад, на рисунку 2.8 показано два можливі набори ваг, які називаються фільтрами (або ядрами згортки).

Перший представлений у вигляді чорного квадрата з вертикальною білою лінією посередині (це матриця 7×7 , яка містить 0, за винятком центрального стовпця, який містить 1); нейрони, які використовують ці вагові коефіцієнти, ігноруватимуть усе у своєму рецептивному полі, крім центральної вертикальної

лінії (оскільки всі вхідні дані будуть помножені на 0, за винятком тих, що розташовані в центральній вертикальній лінії).

Другий фільтр — це чорний квадрат із горизонтальною білою лінією посередині. Знову ж таки, нейрони, які використовують ці ваги, ігноруватимуть усе у своєму рецептивному полі, крім центральної горизонтальної лінії. Тепер, якщо всі нейрони в шарі використовують той самий вертикальний лінійний фільтр (і той самий термін зміщення), і ви подаєте в мережу вхідне зображення, показане на рисунку 2.8 (нижнє зображення), шар виведе верхнє ліве зображення. Зауважте, що вертикальні білі лінії підкреслюються, а решта розмиваються.

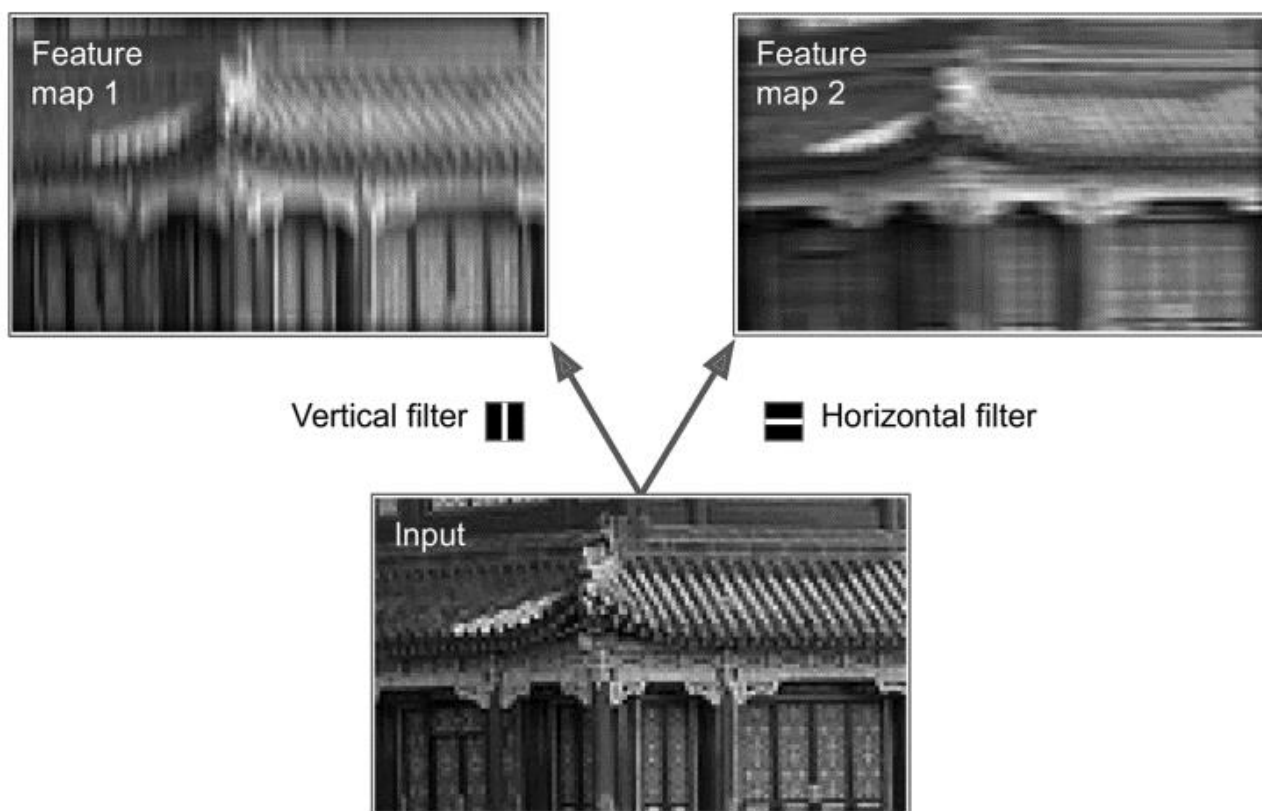


Рисунок 2.8 – Застосування двох різних фільтрів для отримання двох карт функцій

Подібним чином зображення у верхньому правому куті – це те, що ви отримаєте, якщо всі нейрони використовують той самий горизонтальний лінійний

фільтр, зверніть увагу, що горизонтальні білі лінії підкреслюються, а решта розмиваються.

Таким чином, шар, заповнений нейронами, що використовує той самий фільтр, видає карту функцій, яка виділяє області зображення, які найбільше активують фільтр.

Звичайно, вам не потрібно визначати фільтри вручну: натомість під час навчання згортковий шар автоматично вивчатиме найкорисніші фільтри для свого завдання, а шари вище навчатимуться комбінувати їх у складніші шаблони, коли вони бачать цей шаблон, і виводять результат у наступний фільтр. Усі ці карти функцій можна потім створити в кінцевому зображенні та використовувати для його обробки.

Укладання кількох карт функцій

Досі для простоти ми представляли результат кожного згорткового шару як 2D-шар, але насправді згортковий шар має кілька фільтрів (ви вирішуєте, скільки) і виводить одну карту функцій на фільтр, тому він представлений більш точно у 3D на рисунку 2.9. Вона має по одному нейрону на піксель у кожній карті ознак, і всі нейрони в цій карті властивостей мають однакові параметри (тобто однакові ваги та зсув).

Нейрони в різних картах функцій використовують різні параметри. Рецептивне поле нейрона таке ж, як описано раніше, але воно поширюється на всі карти характеристик попередніх шарів. Коротше кажучи, згортковий рівень одночасно застосовує кілька фільтрів, які можна навчати, до своїх вхідних даних, що робить його здатним виявляти кілька функцій будь-де на своїх вхідних даних.

Той факт, що всі нейрони в карті ознак мають однакові параметри, значно зменшує кількість параметрів у моделі. Коли CNN навчиться розпізнавати шаблон в одному місці, він зможе розпізнати його в будь-якому іншому місці.

Навпаки, якщо звичайний DNN навчився розпізнавати шаблон в одному місці, він зможе розпізнати його лише в цьому конкретному місці.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 30 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Вхідні зображення також складаються з кількох підшарів: по одному на колірний канал. Зазвичай їх три: червоний, зелений і синій (RGB). Зображення у відтінках сірого мають лише один канал, але деякі зображення можуть мати набагато більше, наприклад, супутникові зображення, які вловлюють додаткові частоти світла (наприклад, інфрачервоне).

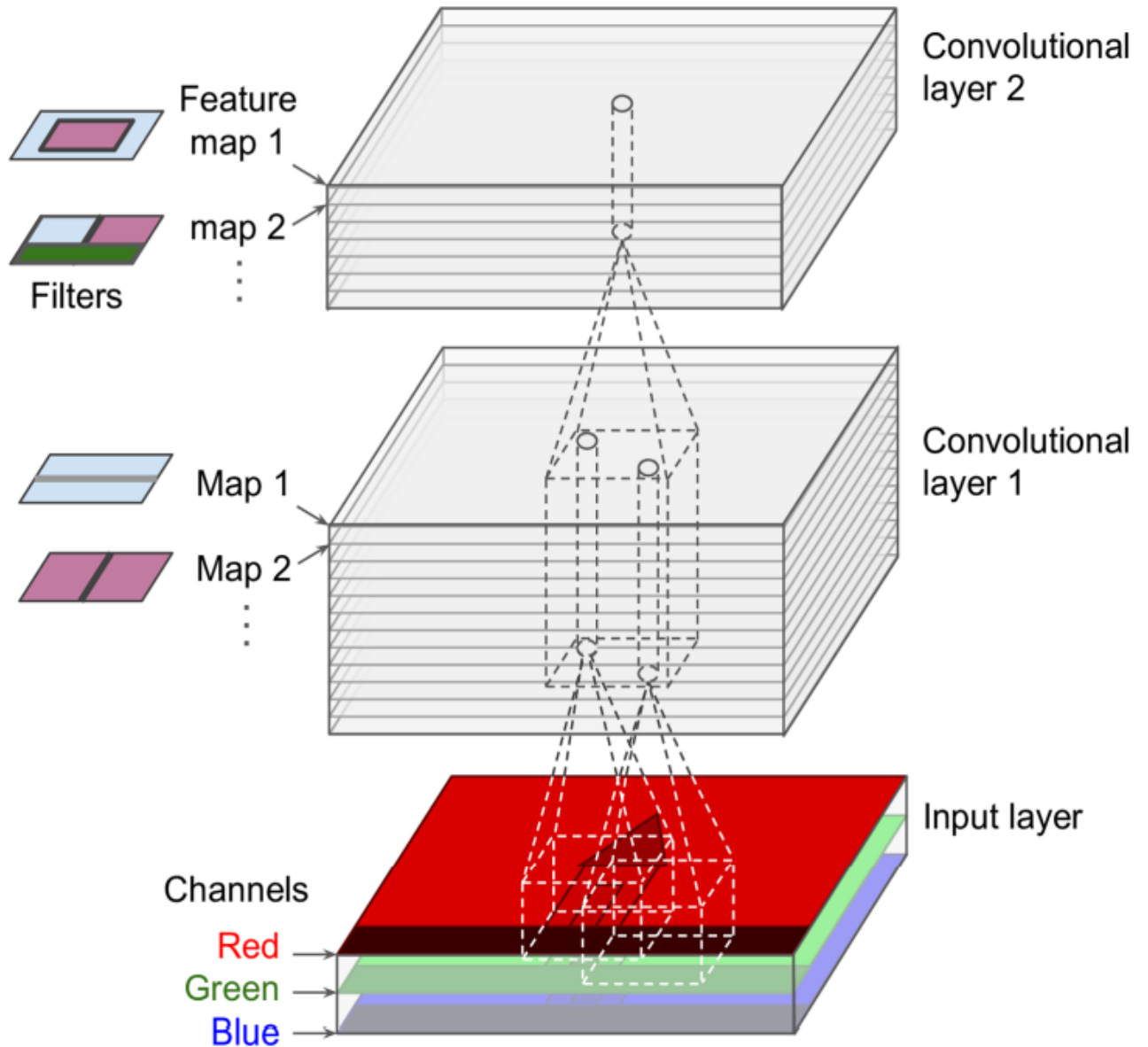


Рисунок 2.9 – Згорткові шари з кількома картами функцій і зображення з трьома кольорові канали

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

Усі нейрони у фільтрі мають однакові ваги, і кожен нейрон в одному фільтрі з'єднаний зі своїм аналогом у наступному фільтрі, так що вихідні дані одного фільтра передаються до відповідного набору нейронів

2.3 Реалізація TensorFlow

Реалізувати максимальний рівень об'єднання в TensorFlow досить просто. Наступний код створює максимальний рівень об'єднання за допомогою ядра 2×2 . За замовчуванням кроки відповідають розміру ядра, тому цей шар використовуватиме крок 2 (як по горизонталі, так і по вертикалі).

За замовчуванням він використовує «дійсне» відступ (тобто без відступу взагалі): Глибоке комп'ютерне бачення з використанням згорткових нейронних мереж `max_pool=keras.layers.MaxPool2D(pool_size=2)` Щоб створити середній шар об'єднання, просто використовуйте `AvgPool2D` замість `MaxPool2D`.

Як і слід було очікувати, він працює так само, як максимальний рівень об'єднання, за винятком того, що він обчислює середнє значення, а не максимальне. Раніше середні шари об'єднання були дуже популярними, але зараз люди переважно використовують рівні максимального об'єднання, оскільки вони, як правило, працюють краще.

Це може здатися дивним, оскільки обчислення середнього значення зазвичай втрачає менше інформації, ніж обчислення максимального.

Але з іншого боку, максимальне об'єднання зберігає лише найсильніші характеристики, позбавляючись усіх безглузких, тому наступні рівні отримують чистіший сигнал для роботи. Крім того, максимальне об'єднання забезпечує сильнішу інваріантність перекладу, ніж середнє об'єднання, і вимагає трохи менше обчислень. Зауважте, що максимальне об'єднання та об'єднання середніх значень можна виконувати вздовж розміру глибини, а не просторового розміру, хоча це не так часто. Це може дозволити CNN навчитися бути інваріантним до різних функцій.

Наприклад, він міг би вивчати кілька фільтрів, кожен з яких виявляв різне обертання того самого шаблону на рисунку 2.10, а рівень максимального об'єднання по глибині гарантував би, що результат буде однаковим незалежно від обертання. CNN так само міг би навчитися бути інваріантним до будь-чого іншого: товщини, яскравості, перекосу, кольору тощо.

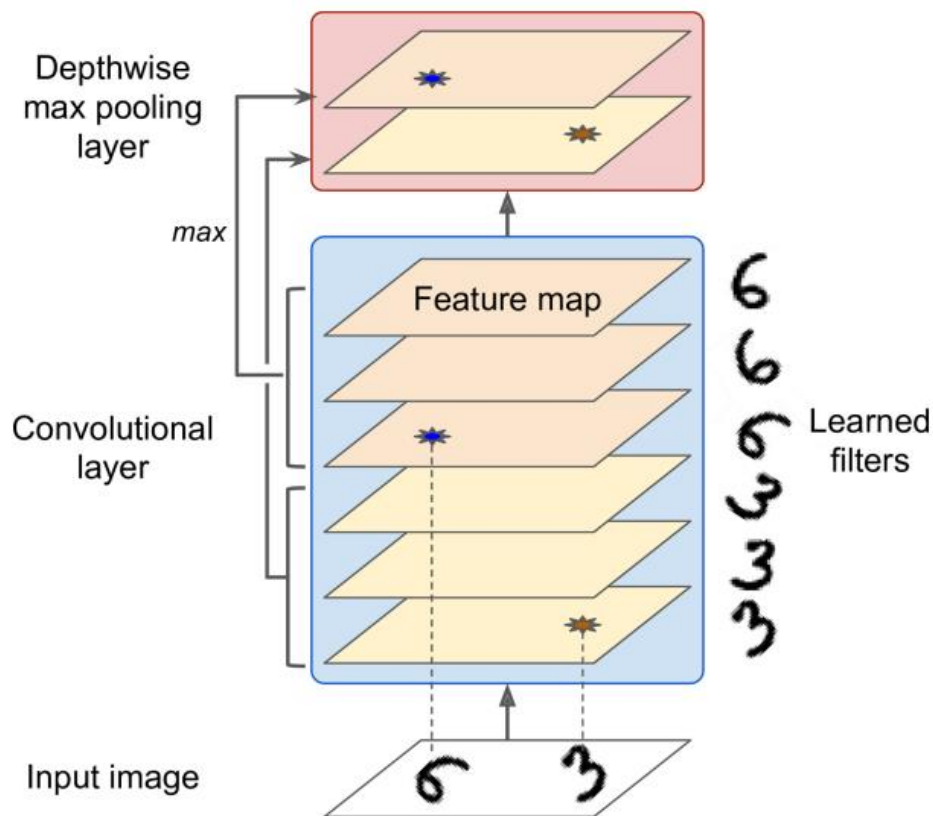


Рисунок 2.10 – Поглиблене максимальне об'єднання, допомога CNN вивчити будь-яку інваріантність

У TensorFlow кожне вхідне зображення зазвичай представляється як тривимірний тензор форми [висота, ширина, канали]. Міні-серія представлена як 4D-тензор форми [розмір міні-партії, висота, ширина, канали]. Ваги згорткового шару представлені надсилається як 4D тензор форми [fh, fw, fn', fn].

Члени зміщення згорткового шару просто представлені як одновимірний тензор форми [fn]. Давайте розглянемо простий приклад.

Наступний код завантажує два зразки зображень за допомогою `load_sample_image()` Scikit-Learn (який завантажує два кольорові зображення, одне з китайським храмом, а інше з квіткою), потім створює два фільтри та застосовує їх до обох зображень, і, нарешті, він відображає одну з отриманих карт функцій.

```
from sklearn.datasets import load_sample_image
# Load sample images
china = load_sample_image("china.jpg") / 255
flower = load_sample_image("flower.jpg") / 255
images = np.array([china, flower])
batch_size, height, width, channels = images.shape
# Create 2 filters
filters = np.zeros(shape=(7, 7, channels, 2), dtype=np.float32)
filters[:, 3, :, 0] = 1 # vertical line
filters[3, :, :, 1] = 1 # horizontal line
outputs = tf.nn.conv2d(images, filters, strides=1, padding="SAME")
plt.imshow(outputs[0, :, :, 1], cmap="gray") # plot 1st image's 2nd feature map
plt.show()
```

Давайте переглянемо цей код:

- інтенсивність пікселів для кожного кольорового каналу представлена у вигляді байтів від 0 до 255, тому ми масштабуємо ці функції, просто поділивши на 255, щоб отримати значення з плаваючою точкою в діапазоні від 0 до 1;
- потім ми створюємо два фільтри 7×7 (один з вертикальною білою лінією посередині, а інший з горизонтальною білою лінією посередині);
- ми застосовуємо їх до обох зображень за допомогою функції `tf.nn.conv2d()`, яка є частиною низькорівневого API Deep Learning TensorFlow. У цьому прикладі ми використовуємо нульовий наповнювач (`padding="SAME"`) і крок один.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 34 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Нарешті, ми малюємо одну з отриманих карт функцій (подібно до верхнього правого зображення на малюнку(31)). Рядок `tf.nn.conv2d()` заслуговує на трохи більше пояснень:

- зображення є вхідним міні-пакетом (4D-тензор, як пояснювалося раніше).
- `filters` — це набір фільтрів для застосування (також 4D-тензор, як пояснювалося раніше);
- кроків дорівнює одному, але це також може бути одновимірний масив із чотирьох елементів, де двома центральними елементами є вертикальний і горизонтальний кроки (`ш` і `мах`);

Перший і останній елементи наразі мають дорівнювати 1. Одного разу до них можна звикнути вкажіть груповий крок (щоб пропустити деякі екземпляри) і крок каналу (щоб пропустити деякі карти функцій або канали попереднього шару).

- відступ має бути «SAME» або «VALID»: — Якщо встановлено значення «SAME», згортковий шар за потреби використовує нульове доповнення. Розмір виходу встановлюється як кількість вхідних нейронів, поділена на крок, округлений у більшу сторону.

Наприклад, якщо вхідний розмір дорівнює 13, а крок – 5 дивлячись на рисунку 2.11), вихідний розмір дорівнює 3 (тобто $13/5 = 2,6$, округлено до 3). Потім нулі додаються якомога рівномірніше навколо входів, якщо це необхідно. Коли `strides=1`, вихідні дані шару матимуть ті самі просторові розміри (ширина та висота), що й входи, отже, така ж назва. Якщо встановлено значення «VALID», згортковий шар не використовує нульове доповнення та може ігнорувати деякі рядки та стовпці внизу та праворуч від вхідного зображення, залежно від кроку, як показано на рисунку 2.11 (для спрощення, лише верхня тут показаний ізонтальний розмір, але, звісно, та сама логіка застосовується довертикальних розмірів).

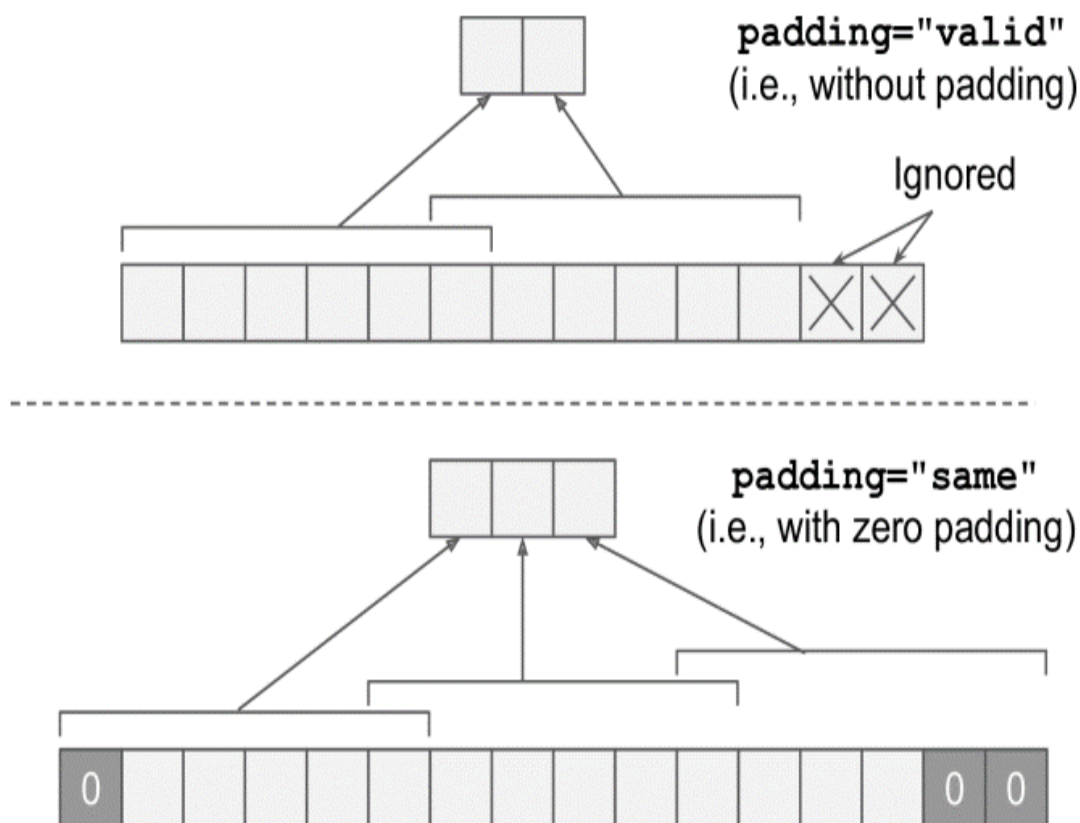


Рисунок 2.11 – Padding="SAME" або "VALID" із шириною введення 13, шириною фільтра 6, кроком

У цьому прикладі було визначено фільтри вручну, але в справжньому CNN ви б зазвичай визначали фільтри як змінні, які можна навчити, щоб нейронна мережа могла дізнатися, які фільтри працюють найкраще, як пояснювалося раніше. Замість того, щоб вручну створювати змінні, використовуйте шар `keras.layers.Conv2D`:

Цей код створює шар `Conv2D` із 32 фільтрами, кожен 3×3 , використовуючи крок перший (по горизонталі та вертикалі) і «однаковий» відступ, а також застосовуючи функцію активації `ReLU` до його виходів. Як бачите, згорткові шари мають чимало гіперпараметрів: ви повинні вибрати кількість фільтрів, їх висоту та ширину, кроки та тип заповнення. Як завжди, ви можете використовувати перехресну перевірку, щоб знайти правильні значення гіперпараметрів, але це забирає багато часу.

2.4 Вимоги до пам'яті та об'єднання шарів

Ще одна проблема з CNN полягає в тому, що згорткові шари потребують величезного обсягу оперативної пам'яті. Це особливо вірно під час навчання, оскільки зворотний прохід зворотного поширення вимагає всіх проміжних значень, обчислених під час проходження вперед.

Наприклад, розглянемо згортковий шар із фільтрами 5×5 , який виводить 200 карт об'єктів розміром 150×100 із кроком 1 і «однаковим» відступом. Якщо вхідним сигналом є зображення RGB розміром 150×100 (три канали), то кількість параметрів становить $(5 \times 5 \times 3 + 1) \times 200 = 15\,200$ (+ 1 відповідає умовам зміщення), що досить мало порівняно з до повністю зв'язаного рівня.

Проте кожна з 200 карт властивостей містить 150×100 нейронів, і кожен із цих нейронів має обчислити зважену суму своїх $5 \times 5 \times 3 = 75$ вхідних даних: це загалом 225 мільйони множень з плаваючою точкою. Не такий поганий, як повністю підключений рівень, але все одно досить інтенсивний з точки зору обчислень.

Крім того, якщо карти функцій представлені за допомогою 32-розрядних чисел з плаваючою точкою, то вихід згорткового шару займатиме $200 \times 150 \times 100 \times 32 = 96$ мільйонів біт (12 Мбайт) оперативної пам'яті.

І це лише для одного випадку — якщо навчання пакет містить 100 екземплярів, тоді цей шар буде використовувати 1,2 ГБ оперативної пам'яті! Під час висновку (тобто під час створення прогнозу для нового екземпляра) оперативна пам'ять, зайнята одним шаром, може бути звільнена, щойно буде обчислено наступний рівень, тому вам потрібно лише стільки оперативної пам'яті, скільки потрібно двом послідовним рівням.

Але під час навчання все, обчислене під час прямого проходження, потрібно зберегти для зворотного проходження, тому необхідний обсяг оперативної пам'яті

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 37 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

дорівнює (принаймні) загальному об'єму оперативної пам'яті, необхідному для всіх рівнів.

Якщо навчання виходить з ладу через помилку браку пам'яті, то можна спробувати зменшити розмір міні-пакета. Крім того, можна спробувати зменшити розмірність за допомогою кроку або видалити кілька шарів. Або можна спробувати використовувати 16-бітні числа з плаваючою точкою замість 32-бітних. Або можна розповсюдити CNN на кількох пристроях.

Об'єднання шарів

Коли ви зрозумієте, як працюють згорткові шари, зрозуміти шари об'єднання буде досить легко. Їх мета полягає в тому, щоб зробити підвибірку (тобто зменшити) вхідне зображення, щоб зменшити обчислювальне навантаження, використання пам'яті та кількість параметрів (таким чином обмежуючи ризик переобладнання). Так само, як у згорткових шарах, кожен нейрон шару об'єднання з'єднаний з виходами обмеженої кількості нейронів попереднього шару, розташованих у невеликому прямокутному рецептивному полі.

Ви повинні визначити його розмір, крок і тип підкладки, як і раніше. Однак об'єднуючий нейрон не має ваг; все, що він робить, це агрегувати вхідні дані за допомогою функції агрегації, такої як максимальне або середнє значення. На рисунку 2.12 показано максимальний рівень об'єднання, який є найпоширенішим типом рівня об'єднання. У цьому прикладі ми використовуємо ядро об'єднання $2 \times 2, 9$ із кроком 2 і без заповнення. Лише максимальне вхідне значення в кожному сприйнятливому полі потрапляє до наступного рівня, тоді як інші вхідні дані скидаються.

Наприклад, у нижньому лівому приймальному полі на рисунку 2.12 вхідними значеннями є 1, 5, 3, 2, тому лише максимальне значення, 5, поширюється на наступний рівень. Через крок, що дорівнює 2, вихідне зображення має половину висоти та половину ширини вхідного зображення (округлено вниз, оскільки ми не використовуємо відступи).

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 38 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

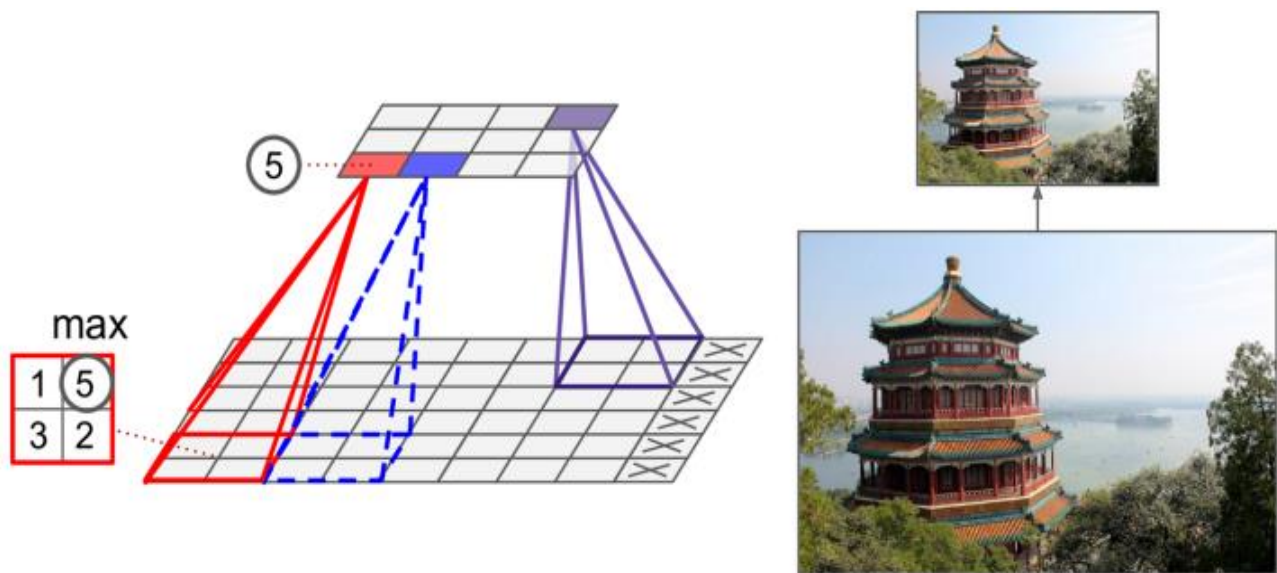


Рисунок 2.12 – Максимальний рівень об'єднання

Рівень об'єднання зазвичай працює на кожному вхідному каналі незалежно, тому глибина виведення є такою самою, як глибина введення. Окрім зменшення обсягу обчислень, використання пам'яті та кількості параметрів, максимальний рівень об'єднання також забезпечує деякий рівень інваріантності для невеликих перекладів, як показано на рисунку 2.13. Тут ми припускаємо, що яскраві пікселі мають нижче значення, ніж темні пікселі, і розглядаємо три зображення (A, B, C), які проходять через максимальний рівень об'єднання з ядром 2×2 і кроком 2. Зображення B і C однакові, як зображення A, але зміщена на один і два пікселі вправо.

Як бачите, результати максимального рівня об'єднання для зображень A і B ідентичні. Ось що означає інваріантність перекладу. Для зображення C вихідні дані відрізняються: воно зсувається на один піксель праворуч (але залишається 75% незмінності).

Вставляючи максимальний рівень об'єднання кожні кілька шарів у CNN, можна отримати певний рівень інваріантності перекладу у більшому масштабі. Крім того, максимальне об'єднання пропонує невелику кількість обертальної інваріантності та незначну інваріантність масштабу. Така інваріантність (навіть

якщо вона обмежена) може бути корисною у випадках, коли прогноз не повинен залежати від цих деталей, наприклад, у завданнях класифікації.

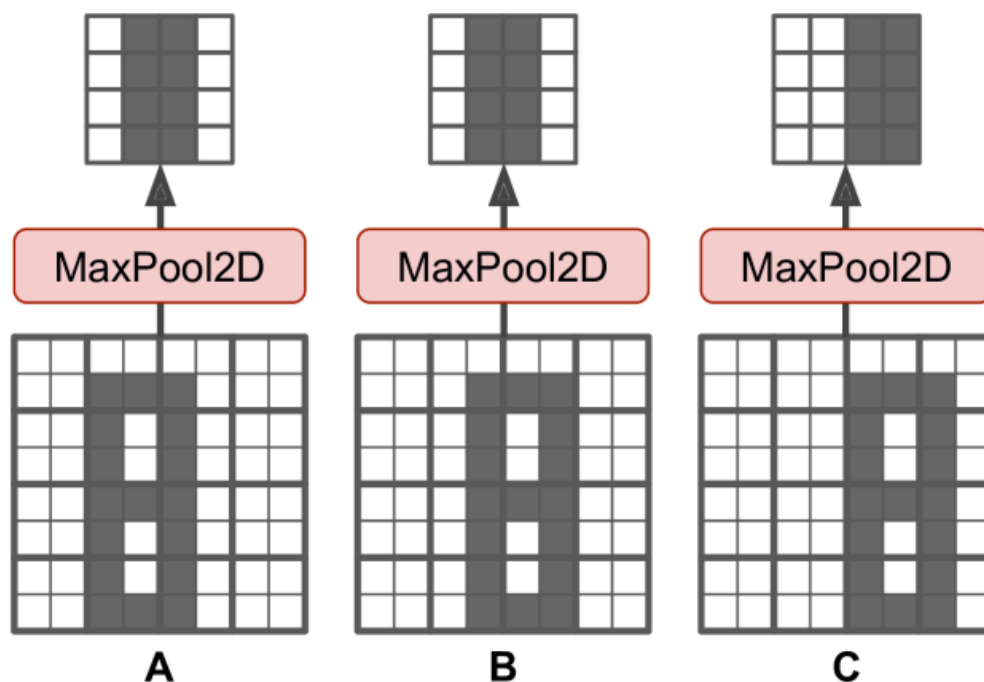


Рисунок 2.13 – Інваріантність до невеликих перекладів

Однак у максимального об'єднання є й недоліки. По-перше, це, очевидно, дуже руйнівний: навіть з крихітним ядром 2×2 і кроком 2, вихід буде вдвічі меншим в обох напрямках (тому його площа буде в чотири рази меншою), просто втрачаючи 75% вхідні значення. А в деяких програмах інваріантність небажана.

Брати семантична сегментація (завдання класифікації кожного пікселя в зображенні відповідно до об'єкта, до якого цей піксель належить, що ми розглянемо пізніше в цій главі): очевидно, якщо вхідне зображення перекладено на один піксель праворуч, вихід повинен також бути транс-на один піксель праворуч. Метою в цьому випадку є еківаріантність, а не інваріантність: невелика зміна вхідних даних повинна призвести до відповідної невеликої зміни на виході.

2.5 Архітектури CNN

Типові архітектури CNN складаються з кількох згорткових шарів (за кожним із яких зазвичай йде шар ReLU (Rectified Linear Unit) - це вид активаційної функції, який часто використовується в нейронних мережах, включаючи нейронні згорткові мережі (CNN)), потім шар об'єднання, потім ще кілька згорткових шарів (+ReLU), потім ще одного рівня об'єднання тощо. Зображення стає все меншим і меншим у міру просування по мережі, але воно також зазвичай стає все глибшим і глибшим (тобто з більшою кількістю карт функцій) завдяки згортковим шарам на рисунку 2.14. У верхній частині стека додається звичайна нейронна мережа прямого зв'язку, що складається з кількох повністю зв'язаних рівнів (+ReLU), а останній рівень виводить прогноз (наприклад, шар softmax, який виводить оцінені ймовірності класу).

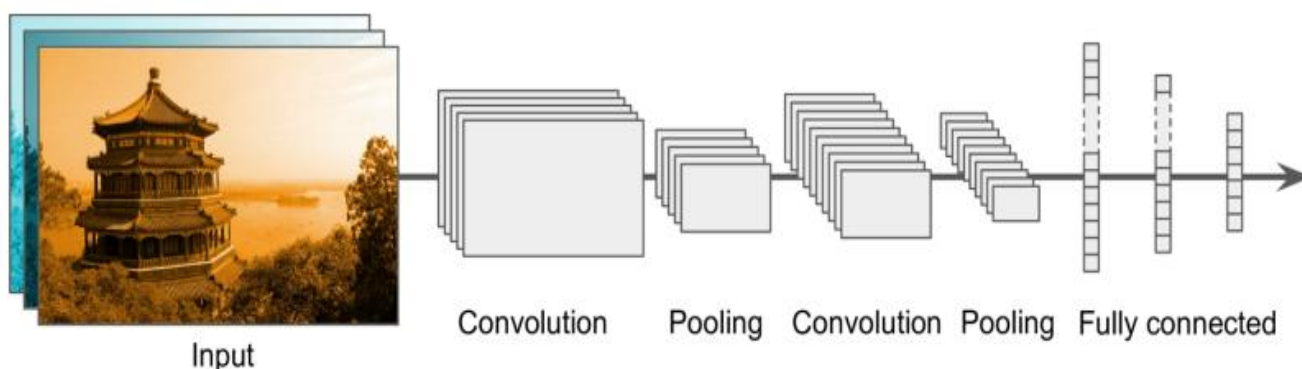


Рисунок 2.14 – Типова CNN архітектура

Згортковий рівень є найважливішим будівельним блоком архітектури CNN, який складається з кількох рівнів залежно від складності розв'язуваної задачі.

Перший шар шару згортки не пов'язаний безпосередньо з кожним пікселем у вхідному зображенні, а лише підключений до сприйнятливих функцій вхідного зображення.

Перший шар згорткового шару призначений для захоплення базових (низькорівневих функцій), таких як кольори, лінії, краї тощо. Крім того, існує рівень об'єднання, який працює шляхом зменшення розміру вхідного зображення зі згорткового шару до зробити обчислення більш ефективними.

І Повністю зв'язаний рівень, який використовується для з'єднання кожного нейрона одного шару з усіма нейронами іншого шару.

Поширеною помилкою в CNN є використання надто великих ядер згортки. Наприклад, замість використання згорткового рівня з ядром 5×5 об'єднайте два шари з ядрами 3×3 : він використовуватиме менше параметрів і вимагатиме менше обчислень, і зазвичай працюватиме краще.

Єдиний виняток для першого згорткового шару: він, як правило, може мати велике ядро (наприклад, 5×5), зазвичай із кроком 2 або більше: це зменшить просторовий розмір зображення без втрати зайвої інформації, і оскільки вхідне зображення загалом має лише три канали, це не буде надто дорогим.

Ось як ви можете реалізувати простий CNN для роботи з набором даних Fashion MNIST

```
model = keras.models.Sequential([
keras.layers.Conv2D(64, 7, activation="relu", padding="same",
input_shape=[28, 28, 1]),
keras.layers.MaxPooling2D(2),
keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
keras.layers.MaxPooling2D(2),
keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
keras.layers.MaxPooling2D(2),
keras.layers.Flatten(),
keras.layers.Dense(128, activation="relu"),
```

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 42 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

```
keras.layers.Dropout(0.5),
keras.layers.Dense(64, activation="relu"),
keras.layers.Dropout(0.5),
keras.layers.Dense(10, activation="softmax")
])
```

Давайте розглянемо цю модель:

- перший шар використовує 64 досить великі фільтри (7×7), але без кроку, оскільки вхідні зображення не дуже великі. Він також встановлює `input_shape=[28, 28, 1]`, оскільки зображення мають розмір 28×28 пікселів з одним кольоровим каналом (тобто градаціями сірого);

- далі у нас є максимальний рівень об'єднання, який використовує розмір пулу 2, тому кожен просторовий вимір ділиться на коефіцієнт 2;

- потім ми двічі повторюємо ту саму структуру: два згорткових шари, за якими йде шар максимального об'єднання. Для більших зображень ми можемо повторити цю структуру ще кілька разів (кількість повторень є гіперпараметром, який можна налаштувати);

- далі йде повністю підключена мережа, що складається з двох прихованих щільних шарів і щільного вихідного рівня. Зверніть увагу, що ми повинні зрівняти його вхідні дані, оскільки щільна мережа очікує одновимірний масиву функцій для кожного екземпляра. Ми також додаємо два шари випадання з відсотком випадання 50% кожен, щоб зменшити переобладнання. Частота помилок у першій п'ятірці – це кількість тестових зображень, для яких у перших п'яти передбаченнях системи не було правильної відповіді. Зображення великі (256 пікселів у висоту) і містять 1000 класів, деякі з яких дуже тонкі (спробуйте розрізнити 120 порід собак). Дивлячись на еволюцію переможців, це хороший спосіб зрозуміти, як працюють CNN.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 43 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

2.6 Висновки

У другому розділі дипломної роботи було досліджено методи та аналіз роботи згортальних нейромереж, зокрема розглянуто початкові етапи роботи, згортальних нейромереж, чому вони працювали так, а не інакше, їх принцип роботи та перші виявлення об'єкта на зображенні. Також було пояснено як реалізувати згортальні шари на платформі TensorFlow, вимоги до пам'яті та об'єднання шарів, а також різні архітектури CNN.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 44 |

3 ВИКОРИСТАННЯ ЗГОРТАЛЬНИХ НЕЙРОМЕРЕЖ ДЛЯ БЕЗПЕКИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ФОТО ТА ВІДЕО ПОТОЦІ

3.1 Розпізнавання образів та її робота

Розпізнавання образів — це важливе завдання комп'ютерного зору, яке використовується для виявлення екземплярів візуальних об'єктів певних класів (наприклад, людей, тварин, автомобілів, будівель та територій) у цифрових зображеннях, таких як фотографії чи відеокадри. Метою виявлення об'єктів є розробка обчислювальних моделей, які надають найбільш фундаментальну інформацію, необхідну програмам комп'ютерного зору.

Робота розпізнавання образів:

Розпізнавання образів може виконуватися з використанням традиційних (1) методів обробки зображень, або (2) мереж глибокого навчання.

1) Методи обробки зображень, як правило, не вимагають історичних даних для навчання та за своєю природою неконтрольовані. OpenCV – популярний інструмент для задач обробки зображень.

OpenCV (Open Source Computer Vision Library) – це бібліотека комп'ютерного зору та обробки зображень з відкритим вихідним кодом. Вона надає широкий спектр функцій для роботи із зображеннями та відео, включаючи завантаження та збереження зображень, виявлення об'єктів, сегментацію, розпізнавання облич, а також безліч інших завдань комп'ютерного зору.

2) Методи глибокого навчання зазвичай залежить від контрольованого чи неконтрольованого навчання, у своїй контрольовані методи є стандартом у завданнях комп'ютерного зору. Продуктивність обмежена обчислювальною потужністю графічних процесорів, що стрімко зростає з кожним роком.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КвРКІ 200231.20.02.07 ПЗ | Арк. 45 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

3.2 Плюси і мінуси методів обробки зображень та глибокого навчання

Методи обробки зображень:

Плюси: отже, ці завдання не вимагають анотованих зображень, де люди маркували дані вручну (для контрольованого навчання).

Мінуси: ці методи обмежені кількома факторами, такими як складні сценарії (без одноколірного фону), оклюзія (частково приховані об'єкти), освітлення та тіні та ефект безладу.

Методи глибокого навчання:

Плюси: Виявлення об'єктів за допомогою глибокого навчання значно стійкіше до оклюзії, складних сцен та складного освітлення.

Мінуси: потрібна велика кількість навчальних даних; процес інструкції зображень є трудомістким і дорогим. Наприклад, маркування 500 000 зображень для навчання користувача алгоритму виявлення об'єктів глибокого навчання вважається невеликим набором даних. Однак багато еталонних наборів даних (MS COCO, Caltech, KITTI, PASCAL VOC, V5) забезпечують доступність позначених даних.

3.3 Використання розпізнавання образів для застосування та їх обробки

Розпізнавання образів використовують в таких сферах як:

- розпізнавання осіб та людей;
- інтелектуальна відеоаналітика;
- автономні транспортні засоби;
- інтелектуальна відеохірургія;
- перевірка дефектів;
- виявлення пішоходів;
- AI-навігація дронів.

Розпізнавання осіб та людей

Більшість систем розпізнавання осіб ґрунтуються на розпізнаванні об'єктів. Його можна використовувати для виявлення осіб, класифікації емоцій або виразів та подачі отриманого поля до системи пошуку зображень для ідентифікації конкретної людини з групи.

Виявлення обличчя - один з найпопулярніших варіантів використання виявлення об'єктів, і ви, ймовірно, вже використовуєте його щоразу, коли розблоковує телефон своїм обличчям.

Виявлення людей також зазвичай використовується для розрахунку кількості людей у роздрібних магазинах або забезпечення показників соціального дистанціювання.

Інтелектуальна відеоаналітика

Виявлення об'єктів використовується в інтелектуальній відеоаналітики (IVA) скрізь, де в торгових точках є камери відеоспостереження, щоб зрозуміти, як покупці взаємодіють із продуктами. Ці відеопотоки проходять через конвеєр анонімізації, щоб розмити обличчя людей та знеособити їх. Деякі варіанти використання IVA зберігають конфіденційність, дивлячись тільки на взуття людей, розміщуючи камери нижче рівня колін і гарантуючи, що система фіксує присутність людини без необхідності безпосередньо дивитися на її риси, що ідентифікуються. IVA часто використовується на заводах, в аеропортах та транспортних вузлах для відстеження довжини черг та доступу до зони обмеженого доступу.

Автономні транспортні засоби

Безпілотні автомобілі використовують виявлення об'єктів, щоб виявляти пішоходів, інші автомобілі та перешкоди на дорозі, щоб безпечно пересуватися. Автономні транспортні засоби, оснащені LIDAR, іноді використовують 3D-виявлення об'єктів, коли навколо об'єктів застосовуються прямокутні форми.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КвРКІ 200231.20.02.07 ПЗ | Арк. 47 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

LIDAR (Light Detection and Ranging) – це метод дистанційного зондування, що використовує лазерні промені для вимірювання відстані до об'єктів чи поверхонь

Принцип роботи LIDAR заснований на відправленні лазерного імпульсу та вимірі часу, за який відбитий сигнал повертається назад до датчика. Шляхом аналізу часу затримки між відправкою імпульсу та прийомом відбитого сигналу можна визначити відстань до об'єктів.

Інтелектуальна відеохірургія

Хірургічне відео – це дуже зашумлені дані, що знімаються з ендоскопів під час відповідальних операцій. Виявлення об'єктів можна використовувати для виявлення об'єктів, що важко помітити, таких як поліпи або поразки, які вимагають негайного втручання хірурга. Він також використовується для інформування персоналу лікарні про статус операції.

Перевірка дефектів

Компанії-виробники можуть використовувати виявлення об'єктів виявлення дефектів на виробничій лінії. Нейронні мережі можна навчити виявляти найдрібніші дефекти, від складок на тканині до вм'ятин чи спалахів у ливарних пластмасах.

На відміну від традиційних підходів до машинного навчання, виявлення об'єктів на основі глибокого навчання також може виявляти дефекти в об'єктах, що сильно відрізняються, таких як продукти харчування.

Виявлення пішоходів

Це одне з найважливіших завдань комп'ютерного зору, що застосовується у робототехніці, відеоспостереженні та автомобільній безпеці. Виявлення пішоходів відіграє ключову роль у дослідженнях виявлення об'єктів, оскільки надає фундаментальну інформацію для семантичного розуміння відеоматеріалів.

Однак незважаючи на відносно високу продуктивність, ця технологія, як і раніше, стикається з такими проблемами, як різні стилі одягу на вигляд або наявність аксесуарів, що закривають, що знижують точність існуючих детекторів.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КвРКІ 200231.20.02.07 ПЗ | Арк. 48 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

AI-навігація дрону

У наші дні дрони оснащені неймовірними камерами і можуть використовувати моделі, розміщені у хмарі, для оцінки будь-якого об'єкта, з яким вони стикаються.

Наприклад, їх можна використовувати для огляду важкодоступних ділянок мостів на наявність тріщин та інших структурних пошкоджень або огляду ліній електропередач, замінюючи небезпечні рутинні вертолітні операції.

3.4 Методи розпізнавання образів та їх робота

R-CNN

Перші моделі інтуїтивно починають із пошуку області, а потім виконують класифікацію. В R-CNN метод вибіркового пошуку є альтернативою повному пошуку на зображенні для фіксації розташування об'єкта. Він ініціалізує невеликі області зображення та поєднує їх у ієрархічну групу. Таким чином, остання група є блоком, що містить все зображення. Виявлені області об'єднуються відповідно до різних колірних просторів і показників подібності. Результатом є кілька пропозицій регіонів, які можуть містити об'єкт шляхом злиття невеликих регіонів.

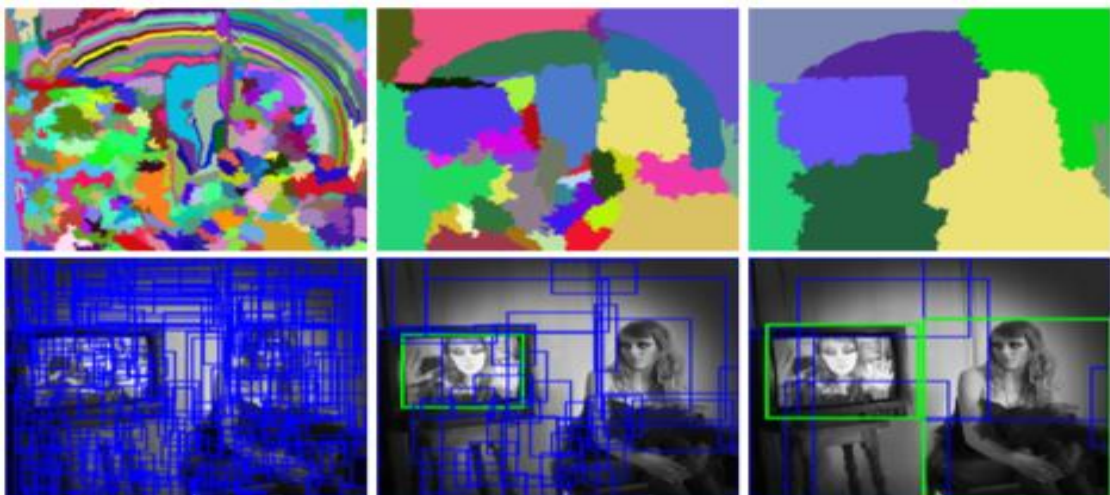


Рисунок 3.1 – Візуалізація результатів сегментації алгоритму, внизу: візуалізація пропозицій області алгоритму

Модель R-CNN поєднує метод вибіркового пошуку для виявлення областей регіонів і глибоке навчання для виявлення об'єкта в цих регіонах. Розмір кожної пропозиції регіону змінюється, щоб відповідати вхідним даним CNN, з яких ми витягуємо вектор ознак із 4096 вимірами. Вектор ознак передається до декількох класифікаторів для отримання ймовірностей приналежності до кожного класу. Кожен із цих класів має класифікатор SVM, навчений робити висновки про можливість виявлення цього об'єкта для заданого вектора ознак. Цей вектор також передає лінійний регресор, щоб адаптувати форми рамки, що обмежує, для пропозиції регіону і, таким чином, зменшити помилки локалізації.

Виявлені області на рисунку 3.2 об'єднуються за різними проміжками кольорів та показниками подібності. Кожна пропозиція регіону передає CNN для отримання вектора ознак, можливі об'єкти виявляються з використанням декількох класифікаторів SVM, а лінійний регресор змінює координати рамки, що обмежує.

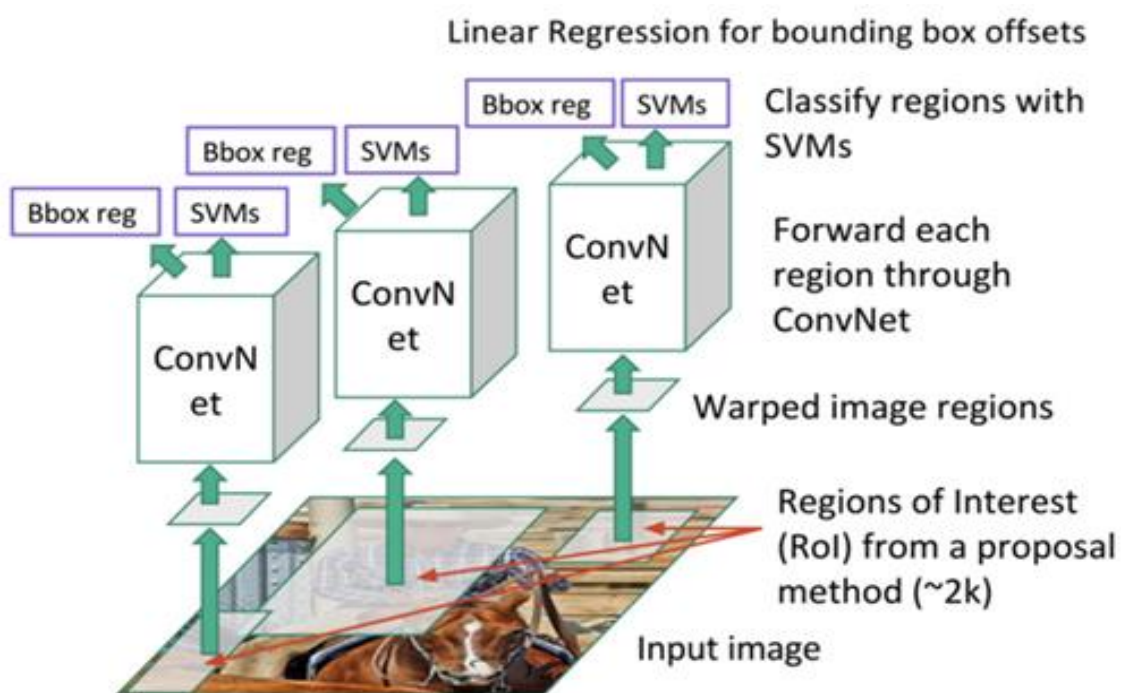


Рисунок 3.2 – Типова структура роботи R-CNN

Кращі моделі R-CNN досягли оцінки mAP 62,4% порівняно з набором тестових даних PASCAL VOC 2012 (збільшення на 22,0 бала порівняно з другим найкращим

результатом у таблиці лідерів) та 31,4% оцінки mAP порівняно з набором даних ImageNet 2013 (збільшення на 7,1 бала порівняно з 2013 роком

SVM (Support Vector Machine) – це метод навчання з учителем, який використовується для завдань класифікації та регресії. Він працює шляхом побудови гіперплощини у багатовимірному просторі, що поділяє точки даних різних класів. Гіперплощина вибирається таким чином, щоб максимізувати відстань між найближчими точками даних різних класів, які називаються опорними векторами.

Мінуси R-CNN

- навчання мережі, як і раніше, займає величезну кількість часу, так як вам доведеться класифікувати 2000 пропозицій регіонів для кожного зображення;
- його не можна реалізувати в режимі реального часу, тому що для кожного тестового зображення потрібно близько 47 секунд;
- алгоритм вибіркового пошуку є фіксованим алгоритмом. Тож цьому етапі навчання немає. Це може призвести до поганих пропозицій регіонів-кандидатів.

Fast R-CNN

Мета швидкої мережі згортання на основі регіонів (Fast R-CNN), полягає в тому, щоб скоротити витрати часу, пов'язані з великою кількістю моделей, необхідних для аналізу всіх пропозицій регіонів.

Основна CNN з кількома шарами згортки приймає все зображення в якості вхідних даних замість використання CNN для кожної пропозиції регіону (R-CNN). Області інтересів (RoI) виявляються за допомогою методу вибіркового пошуку, який застосовується до створених карт об'єктів.

Формально розмір карт об'єктів зменшується з використанням шару пула RoI, щоб отримати допустиму область інтересів з фіксованою висотою та шириною як гіперпараметри. Кожен шар області інтересу передає пов'язані повні шари¹, створюючи вектор ознак. Вектор використовується для прогнозування об'єкта, що спостерігається, за допомогою класифікатора softmax і для адаптації локалізації обмежувальної рамки за допомогою лінійного регресора.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 51 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

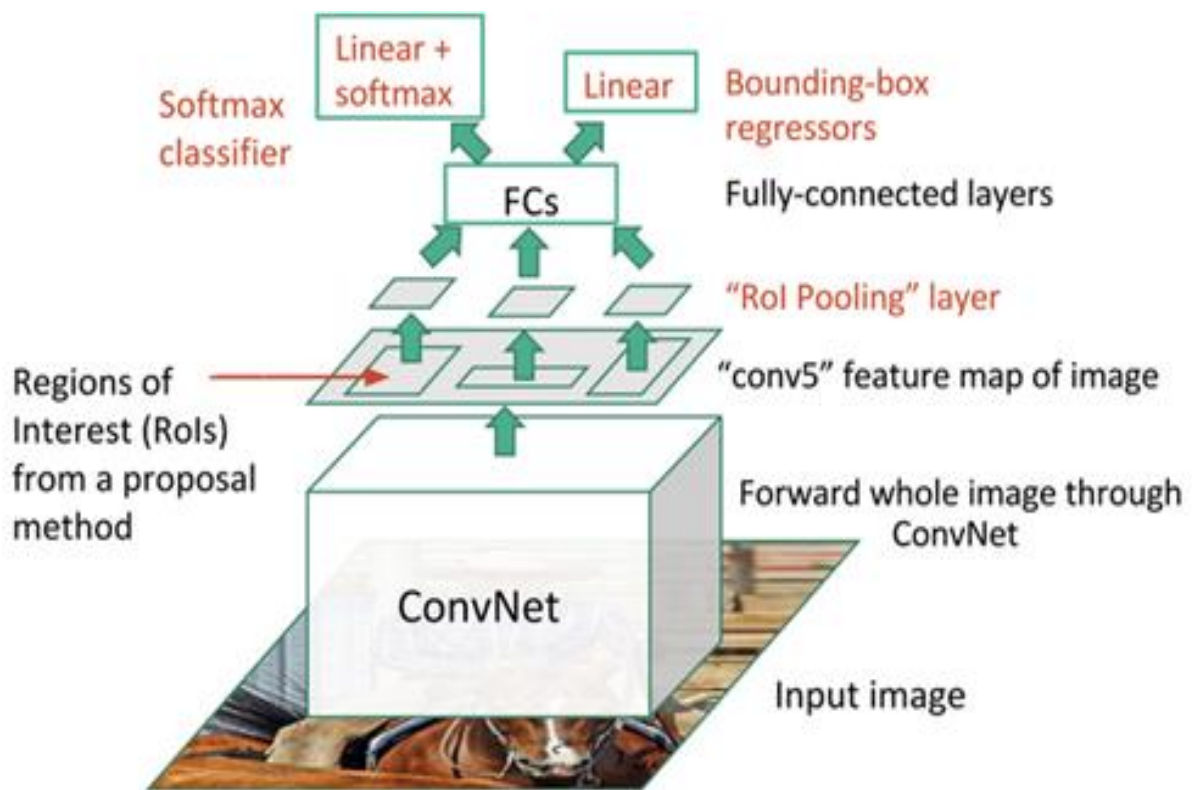


Рисунок 3.3 – Типова структура роботи Fast R–CNN

Кожна область відокремлена за допомогою шару пулу RoI, і він живить пов'язані повністю шари. Цей вектор використовується класифікатором softmax для виявлення об'єкта та лінійним регресором для зміни координат рамки, що обмежує.

Кращі Fast R-CNN досягли оцінки mAp 70,0% для набору тестових даних PASCAL VOC 2007 р., 68,8% для набору тестових даних PASCAL VOC 2010 р. та 68,4% для набору тестових даних PASCAL VOC 2012 р.

Faster R-CNN

Пропозиції регіонів, виявлені за допомогою методу вибіркового пошуку, як і раніше, були необхідні в попередній моделі, яка вимагала значних обчислювальних ресурсів. 2016 року представили мережу пропозицій регіонів (RPN) для прямого створення пропозицій регіонів, прогнозування обмежувальних рамок та виявлення об'єктів. Швидша мережа сітки на основі регіонів (Faster R-CNN) являє собою комбінацію між RPN і моделлю Fast R-CNN.

Модель CNN приймає як вхідні дані все зображення і створює карти характеристик. Вікно розміром 3×3 ковзає по всіх картах об'єктів і виводить вектор ознак, пов'язаний з двома повністю пов'язаними шарами, одна для блокової регресії та одна для блокової класифікації. Пропозиції кількох регіонів передбачаються повністю пов'язаними верствами. Фіксується максимум k областей, тому вихідні дані шару регресії блоків мають розмір $4k$ (координати блоків, їх висота і ширина), а вихідні дані шару класифікації блоків мають розмір $2k$ (об'єктивність балів щоб виявити об'єкт чи ні в коробці). Пропозиції області k , виявлені ковзним вікном, називаються якорями.

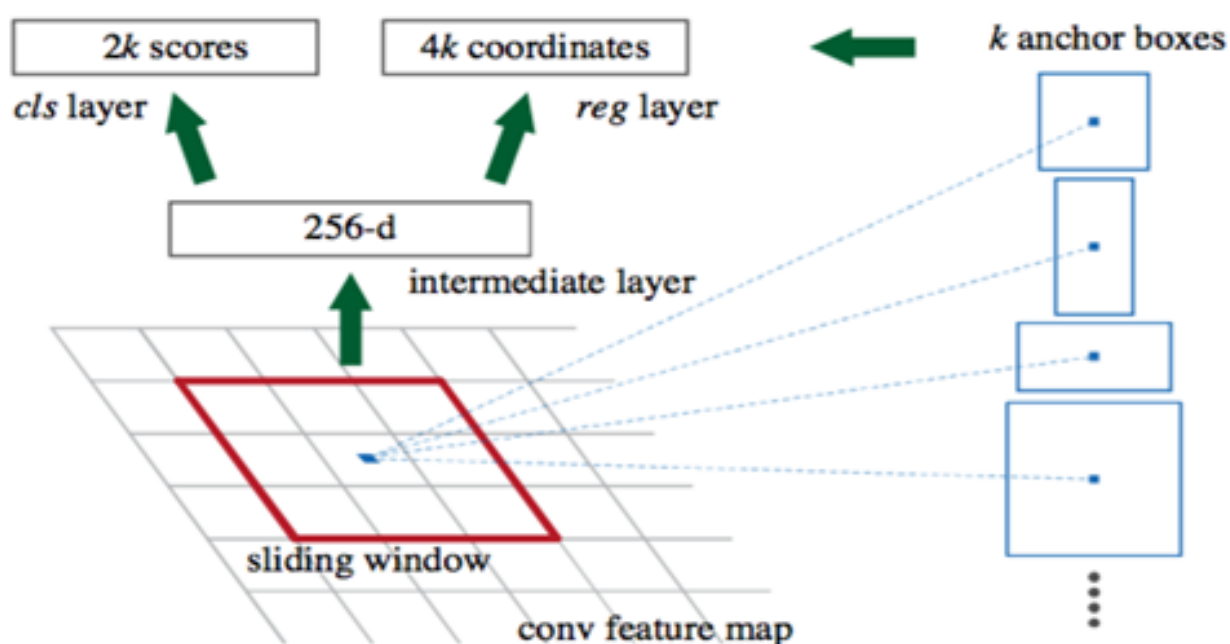


Рисунок 3.4 – Виявлення блоків прив'язки одного вікна 3×3 .

Коли блоки прив'язки виявлені, вони вибираються шляхом застосування порога до показника об'єктивності, щоб залишити тільки відповідні блоки. Ці блоки прив'язки та карти об'єктів, обчислені вихідною моделлю CNN, подають Fast R-CNN на рисунку 3.4.

Вікно розміру 3×3 переміщує всі фільтри ознак та виводить вектор ознак, що передається до двох повністю зв'язаних шарів (Fully connected – FC), один для

корекції регресії та один для класифікації рамок. Передбачення для запропонованих областей відбувається FC шарами.

Faster R-CNN використовує RPN, щоб уникнути методу вибіркового пошуку, прискорити процеси навчання та тестування та підвищити продуктивність.

RPN використовує попередньо вивчену модель набору даних ImageNet для класифікації і точно налаштовує набір даних PASCAL VOC.

Потім згенеровані пропозиції регіонів з якірними полями використовують для навчання Fast R-CNN. Цей процес є ітеративним.

Кожен з цих класів має класифікатор на основі SVM, навчений вивести імовірність виявлення цього об'єкта для певного вектору функцій.

Сформовані вектори надходять у модуль лінійної регресії для коригування форм обмежувальної рамки для пропозиції регіону i , таким чином, зменшує помилку локалізації.

Завдяки Fast-R-CNN можна покращити швидкість виявлення через дві основні зміни: виконання функції виділення ознак зображення перед застосуванням алгоритму пошуку регіонів, таким чином, зображення проходить через CNN лише один раз; заміна SVM на шар softmax, що дозволило розширити можливості нейронної мережі для прогнозування замість створення та тренування окремих SVM-моделей.

Залишалося лише одне проблемне вузьке місце у Fast R-CNN: вибіркового алгоритму пошуку для створення регіонів пропозицій.

Але на заміну вибіркового пошуку прийшла Region Proposal Network (RPN), яка:

- напряму генерувала відеопотоки пропозицій;
- передбачала обмежуючі рамки;
- виявляла об'єкти.

Модель CNN приймає в якості вхідного параметру ціле зображення та створює карти фільтрів (ознак).

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 54 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Її об'єднали в два основних кроки в одну модель, щоб одночасно враховувати виявлення об'єкта (інваріант розташування) та його положення (варіант розташування).

Модель ResNet-101 приймає вихідне зображення як вхідні дані. Останній шар виводить карти об'єктів, кожна з яких спеціалізується на виявленні категорії у будь-якому місці. Наприклад, одна карта ознак спеціалізується на виявленні кішки, інша на банані і так далі. Такі карти об'єктів називаються картами оцінки з урахуванням становища, оскільки враховують просторову локалізацію конкретного об'єкта. Він складається з $k \times k \times (C+1)$ карток оцінок, де k — розмір картки оцінок, а C — кількість класів. Усі ці карти утворюють банк очок. По суті ми створюємо патчі, які можуть розпізнавати частину об'єкта. Наприклад, за $k=3$ ми можемо розпізнати 3×3 частини об'єкта.

Паралельно потрібно запустити RPN для створення області інтересу (RoI). Далі, ми поділяємо кожен банк оцінок на осередки та звіряємо їх із банком результатів. Якщо активовано достатню кількість цих частин, то патч голосує так, як розпізнав об'єкт.

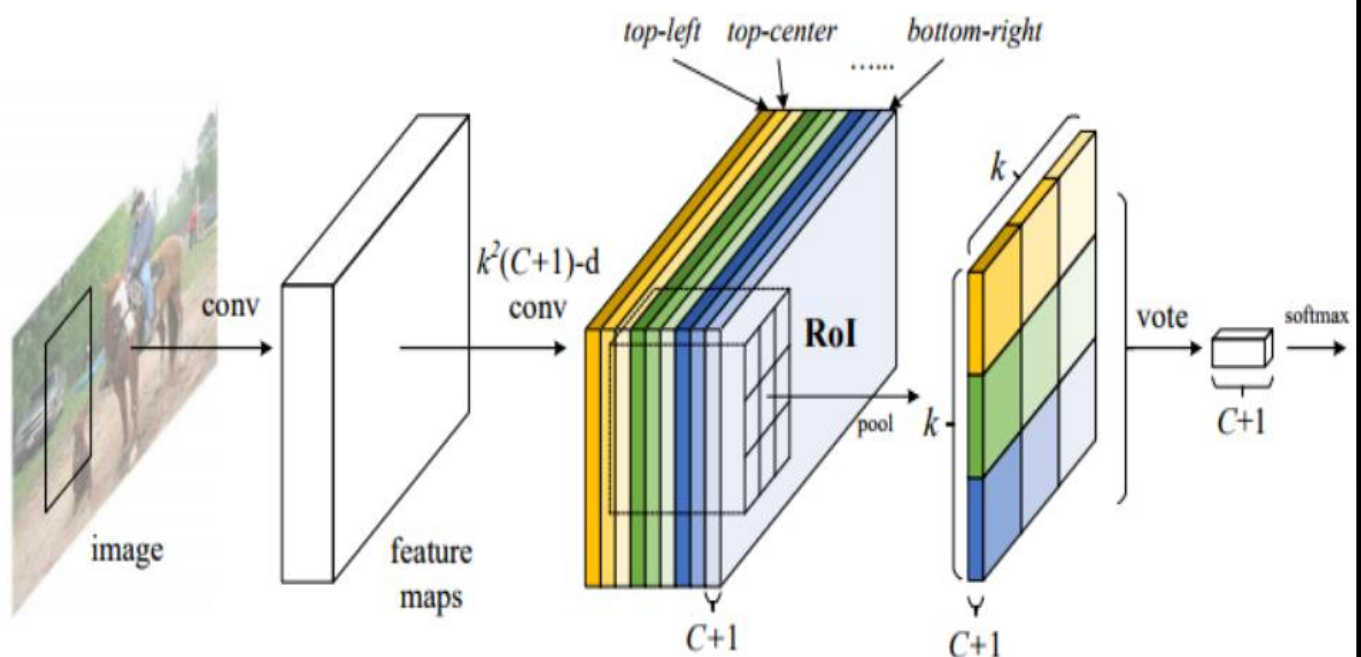


Рисунок 3.6 – Карти балів, чутливі до позиції, і об'єднання RoI

Вхідне зображення на рисунку 3.6 передає модель ResNet для створення карток об'єктів. Модель RPN визначає область інтересів, і кожної області обчислюється оцінка, щоб визначити найбільш ймовірний об'єкт, якщо він є.

R-FCN — це ефективна структура виявлення, яка покращує швидкість FasterRCNN. Конвеєр R-FCN все ще складається з двох альтернатив фази: генерація та класифікація об'єктних пропозицій. Архітектура R-FCN наповнена перевага сучасних повністю згорткових мереж, таких як ResNet, і усунути вузьке місце у швидкості Faster-RCNN пов'язаних рівнів.

На заміну об'єднанню ROI, що не залежить від класу, R-FCN представляє об'єднання ROI, чутливе до позиції, операцію, що залежить від класу та позиції. R-FCN перетворює карти функцій базової мережі до карт оцінок, чутливих до позиції, де кожний канал відповідає певному класу об'єктів і певній позиції об'єднання. Position Sensitive ROI об'єднання виконується поверх вихідних чутливих до позиції балів. рисунок 3.6 демонструє процес об'єднання R-FCN. Така модифікація дозволяє повністю згортковий дизайн для частини класифікації об'єктів інфраструктури R-FCN, що, у свою чергу, значно покращує швидкість роботи. У R-FCN це базова лінія кадру, з якою ми порівнюємо нашу довгострокову структуру навчання представлення для виявлення об'єктів у відео. Ми також використовуємо R-FCN як магістраль архітектури виявлення в цій роботі.

You Only Look Once (YOLO)

Модель YOLO безпосередньо передбачає обмежувальні рамки та ймовірності класів за допомогою однієї мережі в одній оцінці. Простота моделі YOLO дозволяє робити прогнози у реальному часі.

Спочатку модель приймає зображення як вхідні дані. Він поділяє його на сітку $S \times S$. Кожен осередок цієї сітки передбачає B обмежують прямокутників з показником достовірності. Ця впевненість є просто ймовірністю виявлення об'єкта, помножену на IoU між передбаченим і наземним полем істинності на рисунку 3.5.

| Model | mAP | FPS | Real Time speed |
|---------------------|-------|-----|-----------------|
| Fast YOLO | 52.7% | 155 | Yes |
| YOLO | 63.4% | 45 | Yes |
| YOLO VGG-16 | 66.4% | 21 | No |
| Fast R-CNN | 70.0% | 0.5 | No |
| Faster R-CNN VGG-16 | 73.2% | 7 | No |
| Faster R-CNN ZF | 62.1% | 18 | No |

Рисунок 3.9 – Результати порівняних показників YOLO та PASCAL

Плюси YOLO алгоритму:

- швидкість: Цей алгоритм підвищує швидкість виявлення, оскільки може прогнозувати об'єкти у режимі реального часу;
 - висока точність: YOLO - це метод прогнозування, який забезпечує точні результати з мінімальними помилками фону;
 - YOLO може узагальнювати зображення без навантаження пам'яті обробки.
- Мінуси YOLO алгоритму:
- YOLO страждає від значно більшої кількості помилок локалізації та має проблеми з ідентифікацією найближчих предметів.

Single-Shot Detector (SSD)

Подібно до моделі YOLO, розробили одноразовий детектор (SSD) для одночасного прогнозування всіх обмежувальних рамок і ймовірностей класів за допомогою наскрізної архітектури CNN.

отримали 48,5% для IoU = 0,5, 30,3% для IoU = 0,75 і 31,5% для офіційної метрики mAP.

Оскільки використовується бібліотека Open CV алгоритм для детектування зображень може бути замінений на використання загорткової нейромережі. Модуль оптимізації приймає на вході відео потоки зі зображеннями, виявлені на фазі детектування та виконує оптимізації цих зображень засобами Open CV: вирівнювання, поворот, збільшення контрастності, висвітлення та іними. Модуль оптичного розпізнавання тексту (OCR) використовує бібліотеку сучасну система для розпізнавання тексту на основі рекурентних нейромереж типу LSTM (longshort-term memory).

Мінуси методу SSD:

Ступінь точності SSD трохи знижується при ідентифікації дрібніших речей. Якщо модель дуже велика, швидкість може значно впасти.

Mask Region-базований Convolutional Network (Mask R-CNN)

Ще одне розширення моделі Faster R-CNN доданої паралельної гілки до виявлення рамки, що обмежує, щоб передбачити маску об'єкта. Маска об'єкта — це його сегментація пікселів на зображенні. Ця модель перевершує сучасну в чотирьох задачах COCO: сегментація екземпляра, виявлення рамки, що обмежує, виявлення об'єкта і виявлення ключової точки.

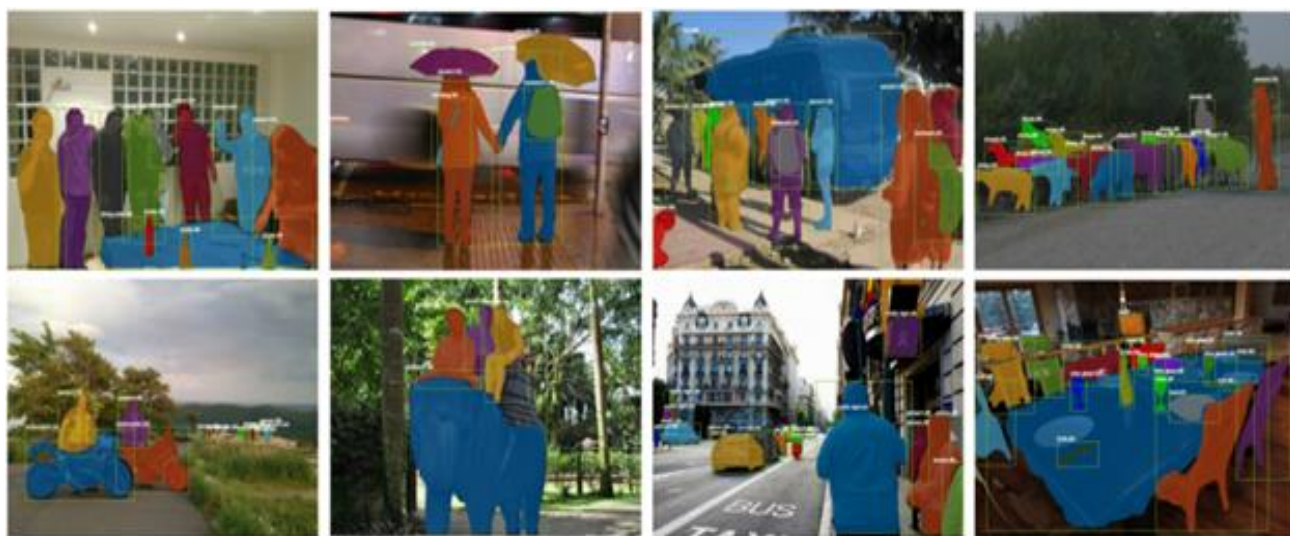


Рисунок 3.12 – Застосування Mask R-CNN у тестовому наборі даних COCO

Модель визначає кожен об'єкт зображення, його локалізацію та точну сегментацію пікселів.

Мережа на основі області маски (Mask R-CNN) використовує швидший конвеєр R-CNN з трьома вихідними гілками для кожного об'єкта-кандидата: мітка класу, зміщення обмежуючої рамки та маска об'єкта. Він використовує мережу регіональних пропозицій (RPN) для створення пропозицій обмежуючої рамки і одночасно створює три результати для кожної області, що цікавить (RoI).

Початковий шар RoIPool, що використовується у Faster R-CNN, замінюється шаром RoIAlign. Він видаляє квантування координат вихідної області інтересу та обчислює точні значення розташування. Шар RoIAlign забезпечує масштабну еквівалентність та трансляційну еквівалентність пропозиціям регіону.

Модель приймає зображення як вхідні дані і передає мережу ResNeXt зі 101 шаром. Ця модель схожа на ResNet, але кожен залишковий блок розрізається на більш легкі перетворення, які поєднуються для додавання розрідженості в блок. Модель виявляє області інтересу, які обробляються за допомогою рівня RoIAlign. Одна гілка мережі пов'язана з повнозв'язним шаром для обчислення координат прямокутників, що обмежують, і ймовірностей, пов'язаних з об'єктами.

Інша гілка пов'язана з двома шарами згортки, останній обчислює маску виявленого об'єкта.

Підсумовуються три функції втрат, пов'язані з кожним розв'язуванням завданням. Ця сума зведена до мінімуму і дає відмінні результати, оскільки розв'язання задачі сегментації покращує локалізацію та, отже, класифікацію.

Mask R-CNN досяг оцінки mAP 62,3% для IoU = 0,5, 43,4% для IoU = 0,7 та 39,8% для офіційної метрики за набором даних COCO test-dev за 2016 рік.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КвРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 64 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

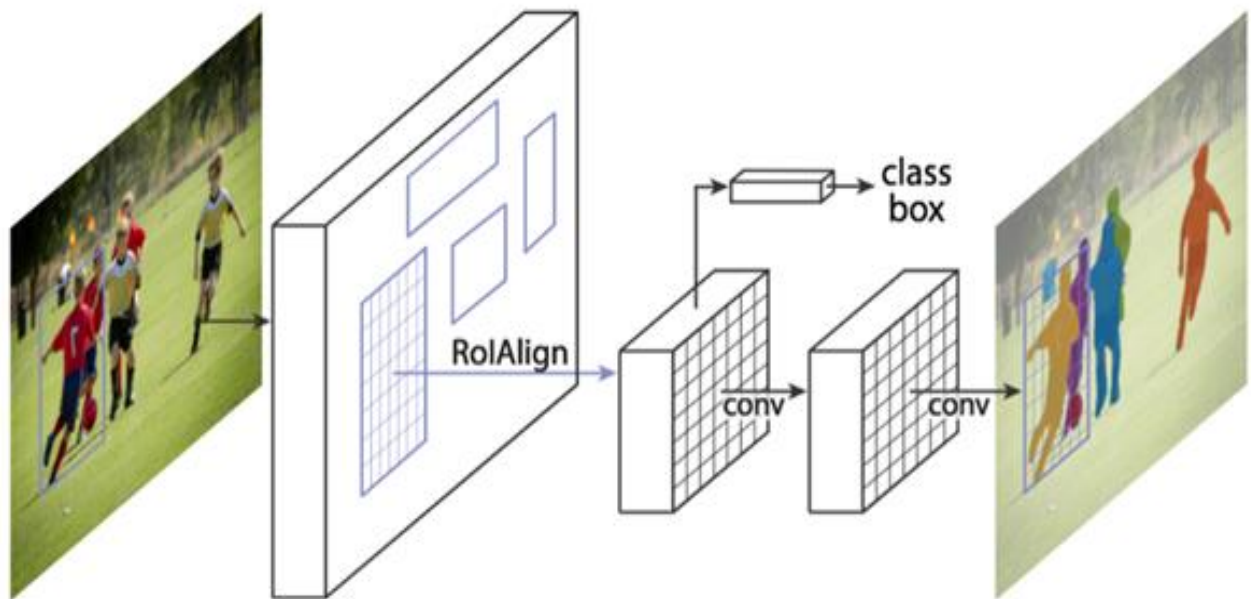


Рисунок 3.13 – Архітектура маски R-CNN

3.5 Зрівняння алгоритмів розпізнавання образів

Найбільш популярним зразком є набір даних Microsoft COCO. Різні моделі зазвичай оцінюються відповідно до показника середньої точності (MAP).

Далі ми порівняємо найкращі алгоритми виявлення об'єктів у реальному часі. Важливо, що вибір алгоритму залежить від варіанта використання та застосування; різні алгоритми чудово справляються з різними завданнями (наприклад, Beta R-CNN показує найкращі результати виявлення пішоходів).

Найкращим алгоритмом виявлення об'єктів у реальному часі в 2022 році є:

- YOLOV7;
- Vision Transformer (ViT);
- Swin i DualSwin;
- PP-YOLOE;
- YOLOR;
- YOLOV4;
- EfficientDet.



Рисунок 3.14 – Графік коефіцієнта точності та швидкості різних алгоритмів

Виявлення об'єктів у реальному часі в тесті COCO: найсучасніше за середньою точністю (AP) Найшвидший алгоритм виявлення об'єктів у реальному часі (час виведення) Крім того, у наборі даних MS COCO важливим показником тесту є час виведення (мс/кадр, чим менше), тим краще) або кадрів за секунду (кадрів за секунду, чим вище, тим краще).

Швидкий прогрес у технології комп'ютерного зору дуже помітний у порівнянні часу логічного висновку.

Грунтуючись на поточному часі виведення (чим менше, тим краще), YOLOV7 досягає 3,5 мс на кадр у порівнянні з YOLOV4 12 мс або популярним YOLOV3 29 мс.

Зверніть увагу на те, як введення YOLO (одноетапний детектор) призвело до значного скорочення часу виведення порівняно з будь-якими раніше встановленими методами, такими як двоетапний метод Mask R-CNN (333 мс).

З технічного погляду досить складно осмислено порівнювати різні архітектури та версії моделей.

І Edge AI стає невід'ємною частиною масштабованих рішень AI, нові алгоритми поставляються з полегшеною версією, оптимізованою для Edge (див. YOLOv7-lite або TensorFlow Lite).

3.6 Висновки та перспективи

Безпека розпізнавання об'єктів у фото та відео потоці, вимагає дослідження сучасних методів машинного навчання, зокрема, згорткових нейронних мереж. Проаналізовано методи та архітектуру згорткових нейромереж для детектування об'єктів. Було прослідковано еволюцію моделей нейромереж та методів, які дозволяють пришвидшити їх роботу та точність розпізнавання починаючи за рахунок SSD, Faster R-CNN, Mask R-CNN, YOLO і RetinaNet. Проведено аналіз функціональності бібліотек для роботи з нейронними мережами, фото та відео потоками, описані їхні переваги та недоліки, обґрунтовано вибір бібліотек для їх використання. Розпізнавання об'єктів, як і раніше, залишається однією з найважливіших сфер застосування для правоохоронних органів. Запропоновані методи може застосуватися правоохоронними органами для автоматичного аналізу транспортних потоків, відслідковування автомобілів порушників чи автомобілів, що знаходяться у розшуку. Застосування згорткових нейромереж з рекурентними шарами і обчисленням на сучасних графічних прискорювачах дозволяють досягти хороших результатів у режимі реального часу при вирішенні задач розпізнавання та ідентифікації зображень. За рахунок застосування операції згортки істотно знижується кількість параметрів, що настроюються в порівнянні з традиційними нейромережами, а чергування згортальних шарів дозволяє вибудувати ієрархію ознак, з досить високою швидкістю і точністю розпізнавання номерних знаків у фото та відео потоці. Використання рекурентних шарів, в свою чергу, дає можливість виявити тимчасові залежності і ознаки, що характеризують критичні ситуації.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. |
| | | | | | | 68 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ВИСНОВОК

У результаті дослідження було виявлено, що аналіз згортальних нейронних мереж і методів, що вони використовують, є важливим етапом у дослідженні областей застосування штучного інтелекту. Дослідження також підтвердило потенціал використання згортальних нейронних мереж для забезпечення безпеки при розпізнаванні об'єктів на фотографіях та в відеопотоці. Порівняльний аналіз алгоритмів методів згортальних нейронних мереж підкреслив їх ефективність та потенційні області застосування. Отже, дане дослідження сприяє розробці ефективних систем розпізнавання зображень на основі згортальних нейромереж, що може мати значний вплив на розвиток штучного інтелекту у різних сферах життя. Об'єктом дослідження є процес розпізнавання зображень, а предметом - методи розпізнавання, засновані на згортальних нейромережах.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | КвРКІ 200231.20.02.07 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 69 |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Згорткові нейронні мережі. URL: <https://itmaster.biz.ua/programming/vision/cnns1.html>
2. EfficientNet. URL: <https://paperswithcode.com/method/efficientnet>
3. Прасолов А. А. Метод підвищення якості відеозображення за допомогою нейронної мережі / Київський політехнічний інститут. URL: <https://ela.kpi.ua/server/api/core/bitstreams/a3328517-fe85-4f18-a8ce-6c29462e47d7/content>
4. EfficientDet. URL: <https://paperswithcode.com/method/efficientdet>
5. EfficientDet: масштабоване та ефективне виявлення об'єктів. URL: https://www.youtube.com/watch?v=WP-PxLxMZH8&ab_channel=%D0%9D%D0%B0%D1%83%D1%87%D0%BD%D1%8B%D0%B5%D1%81%D0%B5%D0%BC%D0%B8%D0%BD%D0%B0%D1%80%D1%8B%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%BC%D1%83%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D1%83
6. SpineNet. URL: <https://paperswithcode.com/method/spinenet>
7. SpineNet: нетрадиційна архітектура backbone-мережі від Google Brain. URL: <https://medium.com/nuances-of-programming/spinenet-%D0%BD%D0%B5%D1%82%D1%80%D0%B0%D0%B4%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F->

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 70 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

<https://arxiv.org/abs/1808.07446>
[D1%83%D1%80%D0%B0-backbone-%D1%81%D0%B5%D1%82%D0%B8-%D0%BE%D1%82-google-brain-6758854287bd](https://arxiv.org/abs/1808.07446)

8. Centernet і Centertrack. URL: <https://blog.xperience.ai/centernet-i-centertrack/>
9. ThunderNet. URL: <https://paperswithcode.com/method/thundernet>
10. ThunderNet: на шляху до виявлення універсальних об'єктів у реальному часі.
URL:
https://www.researchgate.net/publication/332070725_ThunderNet_Towards_Real-time_Generic_Object_Detection
11. Dence Net. URL: https://pytorch.org/hub/pytorch_vision_densenet/
12. DenseNet: Повний посібник. URL:
<https://medium.com/@alejandro.itoaramendia/densenet-a-complete-guide-84fedef21dcc#23b1>
13. DenseNet — щільно пов'язані згорткові мережі. URL:
<https://towardsdatascience.com/paper-review-densenet-densely-connected-convolutional-networks-acf9065dfefb>
14. CSPNet: нова магістраль, яка може покращити навчання CNN. URL:
https://www.researchgate.net/publication/343270535_CSPNet_A_New_Backbone_that_can_Enhance_Learning_Capability_of_CNN
15. CSPDenceNet. URL: <https://paperswithcode.com/method/cspdensenet>
16. CSPNet: міжетапна часткова мережа. URL:
<https://cvinvolution.medium.com/cspnet-cross-stage-partial-network->

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 71 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

[2f1374fb527b](#)

17. DetNASNet. URL: <https://paperswithcode.com/method/detnasnet>

18. DetNASNet. URL: <https://serp.ai/detnasnet/>

19. DetNAS: магістральний пошук для виявлення об'єктів. URL: <https://arxiv.org/pdf/1903.10979>

20. Розпізнавання рукописного тексту з допомогою перцептронної мережі. URL: <https://itmaster.biz.ua/programming/vision/mlp-hand-text.html>

21. Згорткові нейронні мережі. URL: <https://itmaster.biz.ua/programming/vision/cnns1.html>

22. Класифікація рукописних цифр з використанням вертлюжних нейронних мереж на основі 8-бітного дола з плаваючою зап'ятою. URL: https://www.researchgate.net/publication/323429756_Handwritten_Digit_Classification_using_8-bit_Floating_Point_based_Convolutional_Neural_Networks/link/5a95b5530f7e9ba42972ca62/download?tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uRG93bmxvYWQlLCJwYWdlIjoicHVibGljYXRpb24iLCJwcmV2aW91c1BhZ2UiOiJfZGlyZWN0In19

23. Як навчити нейронні мережі класифікації зображень. URL: <https://sandy-lee.medium.com/how-to-train-neural-networks-for-image-classification-part-2-c9935445c562>

24. Декілька типів архітектури штучних нейронних мереж, які вам слід знати.

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 72 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

URL: <https://dedekurniawann.medium.com/several-types-of-artificial-neural-networks-architecture-that-you-should-know-c169a5e22ec7>

25.Карти оцінок з урахуванням позиції та пул RoI з урахуванням позиції. URL:

https://www.researchgate.net/figure/7-Position-sensitive-score-maps-and-Position-sensitive-RoI-Pooling-Dai-et-al-2016_fig20_327800511

26.Застосування згорткових нейронних мереж для безпеки розпізнавання об'єктів у відеопотоці. URL:<https://core.ac.uk/download/pdf/327690899.pdf>

27.Вивчення візуальних моделей для виявлення людей та прогнозування дій.

URL:https://www.researchgate.net/publication/327800511_Learning_visual_models_for_person_detection_and_action_prediction

28.Вивчення та використання моделювання даних. URL:

https://www.researchgate.net/publication/343653793_The_way_to_a_smarter_community_Exploring_and_Exploiting_Data_Modeling_Big_Data_Analytics_High_Performance_Computing_and_Artificial_Intelligence_Techniques_for_Applications_of_2D_Energy_-_Dispersive_Det

29.Рисунок 1.1. URL: <https://paperswithcode.com/method/efficientnet>

30.Рисунок 1.2. URL: <https://ar5iv.labs.arxiv.org/html/1905.11946>

31.Рисунок 1.3. URL: <https://paperswithcode.com/method/efficientdet>

32.Рисунок 1.4 – 1.15 - 1.16. URL: [https://medium.com/nuances-of-programming/spinenet-](https://medium.com/nuances-of-programming/spinenet-%D0%BD%D0%B5%D1%82%D1%80%D0%B0%D0%B4%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F-)

[%D0%BD%D0%B5%D1%82%D1%80%D0%B0%D0%B4%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F-](https://medium.com/nuances-of-programming/spinenet-%D0%BD%D0%B5%D1%82%D1%80%D0%B0%D0%B4%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F-)

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------------|
| | | | | | КВРКІ 200231.20.02.07 ПЗ | Арк. 73 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

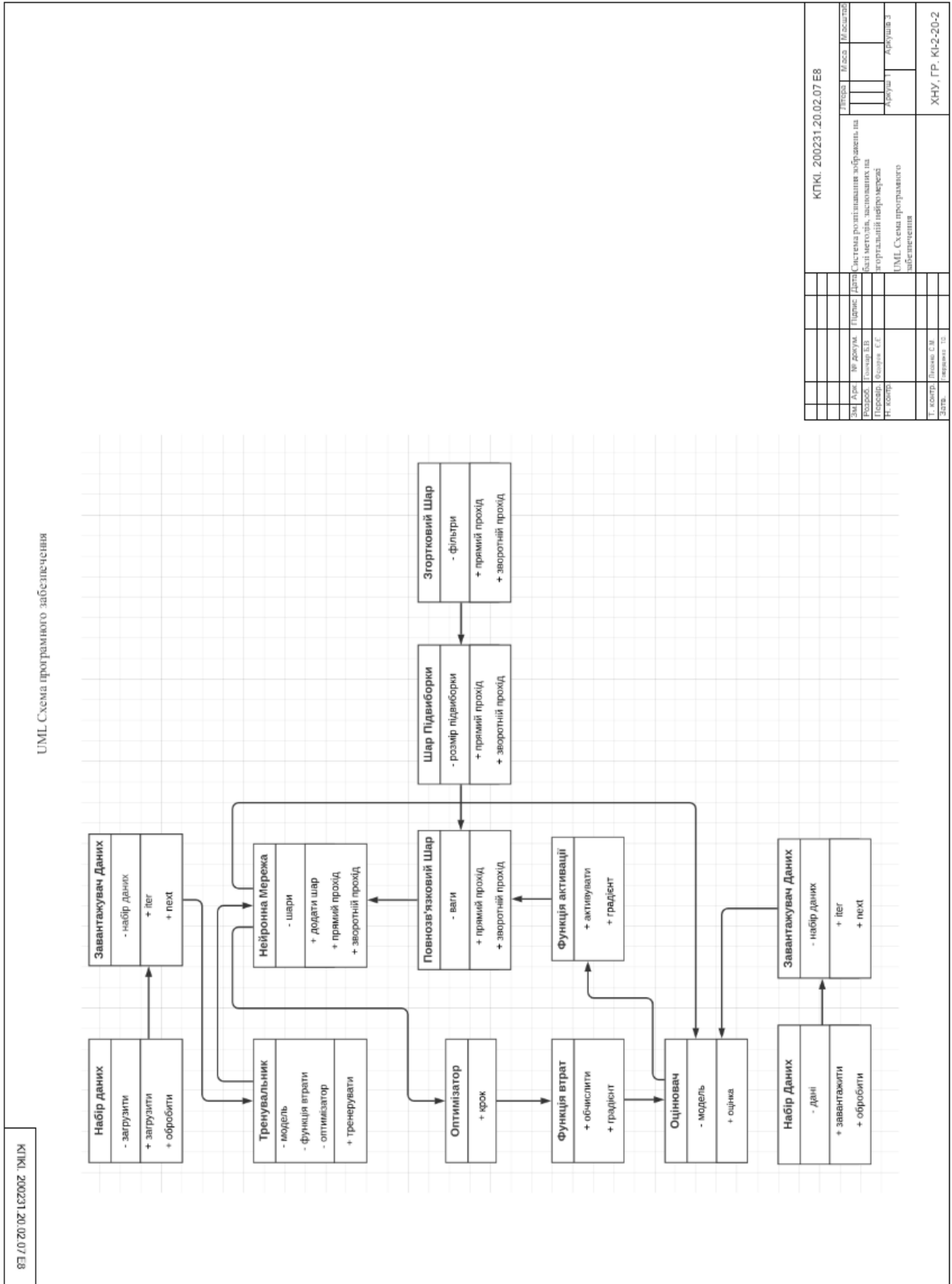
[%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0-backbone-%D1%81%D0%B5%D1%82%D0%B8-%D0%BE%D1%82-google-brain-6758854287bd](https://paperswithcode.com/method/spinenet)

- 33.Рисунок 1.5. URL: <https://paperswithcode.com/method/spinenet>
- 34.Рисунок 1.6. URL: <https://blog.xperience.ai/centernet-i-centertrack/>
- 35.Рисунок 1.7. URL: <https://paperswithcode.com/method/thundernet>
- 36.Рисунок 1.8 - 1.9. URL: <https://medium.com/@alejandro.itoaramendia/densenet-a-complete-guide-84fedef21dcc#23b1>
- 37.Рисунок 1.10 – 1.19. URL: https://www.researchgate.net/publication/343270535_CSPNet_A_New_Backbone_that_can_Enhance_Learning_Capability_of_CNN
- 38.Рисунок 1.11. URL: <https://paperswithcode.com/method/detnasnet>
- 39.Рисунок 1.12. URL: <https://arxiv.org/pdf/1903.10979>
- 40.Рисунок 1.13. URL: <https://ela.kpi.ua/server/api/core/bitstreams/a3328517-fe85-4f18-a8ce-6c29462e47d7/content>
- 41.Рисунок 1.14. URL: <https://blog.ml6.eu/retraining-efficientdet-for-high-accuracy-object-detection-961e906cae8>
- 42.Рисунок 1.17. URL: <https://picture.iczhiku.com/weixin/message1586767492472.html>
- 43.Рисунок 1.18. URL: https://www.researchgate.net/publication/332070725_ThunderNet_Towards_Real-time_Generic_Object_Detection

- 44.Рисунок 1.20. URL: <https://towardsdatascience.com/paper-review-densenet-densely-connected-convolutional-networks-acf9065dfefb>
- 45.Рисунок 2.1 – 2.5 – 2.6 – 2.7 – 2.10 – 2.12 – 2.13. URL: <https://www.kaggle.com/code/viroviro/introduction-to-cnns>
- 46.Рисунок 2.2. URL: <https://congyuzhou.medium.com/lenet-5-%D1%81%D0%B2%D0%BE%D0%B8%D0%BC%D0%B8-%D1%80%D1%83%D0%BA%D0%B0%D0%BC%D0%B8-b60ae3727cd3>
- 47.Рисунок 2.3. URL: <https://itmaster.biz.ua/programming/vision/mlp-hand-text.html>
- 48.Рисунок 2.4. URL: <https://itmaster.biz.ua/programming/vision/cnns1.html>
- 49.Рисунок 2.8 – 2.9. URL: <https://sandy-lee.medium.com/how-to-train-neural-networks-for-image-classification-part-2-c9935445c562>
- 50.Рисунок 2.14. URL: <https://dedekurniawann.medium.com/several-types-of-artificial-neural-networks-architecture-that-you-should-know-c169a5e22ec7>
- 51.Рисунок 3.1 – 3.2 – 3.3 – 3.4 – 3.5 – 3.6 – 3.7 – 3.8 – 3.9 – 3.10 – 3.11 – 3.12 – 3.13 – 3.16. URL: <https://www.artfido.com/the-forest-and-the-trees-deep-learning-and-object-recognition/>
- 52.Рисунок 3.14 – 3.15. URL: <https://viso.ai/deep-learning/object-detection/>

Додаток А (обов'язковий)

Копія креслення «UML Схема програмного забезпечення»



Додаток В

(обов'язковий)

Копія креслення «Код роботи CNN»

КПКи. 200231.20.02.07.Е8

Код роботи CNN

```
model = keras.models.Sequential([
keras.layers.Conv2D(64, 7, activation="relu", padding="same",
input_shape=[28, 28, 1]),
keras.layers.MaxPooling2D(2),
keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
keras.layers.MaxPooling2D(2),
keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
keras.layers.MaxPooling2D(2),
keras.layers.Flatten(),
keras.layers.Dense(128, activation="relu"),
keras.layers.Dropout(0.5),
keras.layers.Dense(64, activation="relu"),
keras.layers.Dropout(0.5),
keras.layers.Dense(10, activation="softmax")
])
```

| | | | | | | | | | |
|--|----------|----------|--------------------|----------|-----------|----------|----------|----------|----------|
| КПКи. 200231.20.02.07.Е8 | | | | | | | | | |
| Знак | Адрес | № докум. | Філія | Дата | Категорія | Маса | Місце | Місце | Місце |
| Розроб. | Виконав. | Перевір. | Відп. за | Відп. за | Відп. за | Відп. за | Відп. за | Відп. за | Відп. за |
| Н. код | В. код | П. код | В. код | В. код | В. код | В. код | В. код | В. код | В. код |
| Система розпізнавання зображень на базі мережі, заснованих на організованій мережі | | | | | | | | | |
| Код роботи CNN | | | | | | | | | |
| Г. розроб. | Листок | Стор. | ХНУ, ГР. КІ-2-20-2 | | | | | | |
| Дата | Розроб. | Відп. | | | | | | | |

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016290867

Дата перевірки:
28.05.2024 18:56:22 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
28.05.2024 19:03:14 EEST

ID користувача:
100005591

Назва документа: Гончар_Система розпізнавання зображень на базі методів, заснованих на згортальній нейр...

Кількість сторінок: 77 Кількість слів: 11883 Кількість символів: 88658 Розмір файлу: 7.85 MB ID файлу: 1016084548

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

15.1% Схожість

Найбільша схожість: 4.01% з Інтернет-джерелом (<https://nupp.edu.ua/uploads/files/0/events/conf/2023/ix-vnpg-ems/zbi..>

15% Джерела з Інтернету

313

Сторінка 79

2.77% Джерела з Бібліотеки

246

Сторінка 84

2.16% Цитат

Цитати

15

Сторінка 85

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

5

Підозріле форматування

19
сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилко в документах: 13%**

| | | | | |
|--|----------|---------|-----------------------------|---------|
| ID: 127536 Назва: БКР Система розпізнавання зображень на базі методів, заснованих на згортальній нейромережі Додано в БД: 2024-05-28 Автора: Б. В. Гончар Керівники: Є. Є. Федоров Консультанти: Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 78676 | 620 | 303 (0%) | 4 (1%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Гончар Богдан Валентинович

Тема: Система розпізнавання зображень на базі методів, заснованих на згортковій нейромережі.

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 69

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка ефективної системи розпізнавання зображень на базі методів, заснованих на згортальній нейромережі.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи:

У першому розділі було детально розглянуто сучасні системи згортальних нейромереж, як вони автоматично використовують фреймворки, беруть за основу другі методи, оптимізуючи їх та використовуючи їх краще. Також було проведено аналіз та опис їх основних характеристик, методів роботи, та проведено порівняння різних підходів до роботи з неординарними кінцевим результатом. У другому розділі дипломної роботи було досліджено методи та аналіз роботи згортальних нейромереж, зокрема розглянуто початкові етапи роботи, згортальних нейромереж, чому вони працювали так, а не інакше, їх принцип роботи та перші виявлення об'єкта на зображенні. Також було пояснено як реалізувати згортальні шари на платформі TensorFlow, вимоги до пам'яті та об'єднання шарів, а також різні архітектури CNN. У третьому розділі в результаті дослідження було виявлено, що аналіз згортальних нейронних мереж і методів, що вони використовують, є важливим етапом у дослідженні областей застосування штучного інтелекту. Дослідження також підтвердило потенціал використання згортальних нейронних мереж для забезпечення безпеки при розпізнаванні об'єктів на фотографіях та в відеопотоці.

Порівняльний аналіз алгоритмів методів згортальних нейронних мереж підкреслив їх ефективність та потенційні області застосування. Отже, дане дослідження сприяє розробці ефективних систем розпізнавання зображень на основі згортальних нейромереж, що може мати значний вплив на розвиток штучного інтелекту у різних сферах життя. Об'єктом дослідження є процес розпізнавання зображень, а предметом - методи розпізнавання, засновані на згортальних нейромережах.

4. Позитивні сторони роботи: система може детектувати об'єкти на зображеннях, а також вона дозволяє автоматизувати цей процес.

5. Негативні сторони роботи: Потрібно провести більш детальне дослідження згортальних нейромереж з детектування об'єктів.

Потрібно навести більше прикладів промислових систем розпізнавання зображень з детектуванням об'єктів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження:

Треба провести дослідження на більшій кількості датасетів.

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Чешун Віктор Миколайович, к.т.н., доцент

кафедри кібербезпеки

"7" червня 2024 р.

 (підпис)

Завідувачу кафедри КІІС
д-р.техн.наук, проф. Говорущенко Т. О.

Гончара Богдана Валентиновича

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-20-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система розпізнавання зображень на базі методів, заснованих на згортковій нейромережі

Автор: Гончар Богдан Валентинович

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Федоров Євген Євгенійович, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

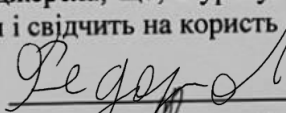
Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 246 джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 15.1% і адресується до 313 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



Є. Є. Федоров

Гарант ОП



Є.М. Лисенко

Завідувач кафедри КПС



Т. О. Говорушенко