УДК 004.932.4

І. О. Бабійчук
О.С. Городецька

ПОРІВНЯЛЬНИЙ АНАЛІЗ АРХІТЕКТУР ДЛЯ ВИЯВЛЕННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ

Вінницький національний технічний університет

Анотація
В роботі здійснено порівняння ефективності архітектур InceptionResNetV2 та YOLOv5 у виявленні об'єктів на зображеннях. Дослідження включає аналіз алгоритмів та практичне тестування моделей на реальних даних.
Ключові слова: Архітектури, виявлення об'єктів, алгоритми, моделі.

Abstract
The study compared the efficiency of the InceptionResNetV2 and YOLOv5 architectures in object detection on images. The research includes an analysis of the algorithms and practical testing of the models on real data.
Keywords: Architectures, object detection, algorithms, models.

Вступ

За останні десятиліття значний прогрес досягнуто в галузі комп'ютерного зору та обробки зображень завдяки розвитку глибоких нейронних мереж. Виявлення об'єктів на зображеннях є однією з ключових задач у цій області, яка знаходить широке застосування у великому ряді індустрій, включаючи медицину, автомобільну промисловість, безпеку та багато інших. Дві із найбільш відомих та ефективних архітектури для виявлення об'єктів на зображеннях - InceptionResNetV2 та YOLOv5 - привертають значну увагу дослідників та практиків у галузі комп'ютерного зору.

Метою даної роботи є порівняння архітектур InceptionResNetV2 та YOLOv5 з точки зору їхньої ефективності та точності у виявленні об'єктів на зображеннях. Дослідження передбачає аналіз алгоритмів, які лежать в основі цих архітектур, порівняння їхніх характеристик та особливостей, а також практичну перевірку результатів роботи моделей на реальних наборах даних. Отримані висновки дозволять визначити, яка з архітектур є більш ефективною та точною у виявленні об'єктів на зображеннях, що є важливим кроком у розвитку комп'ютерного зору та його застосування на практиці.

Загальні відомості

Перш за все, ознайомимось з деякими ключовими термінами, щоб мати уявлення про принципи функціонування, які використовуються в архітектурах InceptionResNetV2 та YOLOv5.

Глибоке навчання - це галузь машинного навчання, яка використовує нейронні мережі з великою кількістю шарів (глибокі моделі) для розпізнавання закономірностей у вхідних даних. Цей підхід дозволяє моделям автоматично вивчати представлення даних на різних рівнях абстракції, що дозволяє їм вирішувати складні завдання класифікації та прогнозування.

Класифікація об'єктів в глибокому навчанні - це процес призначення класу або мітки до вхідного зображення. Наприклад, якщо маємо зображення автомобіля, система класифікації об'єктів може визначити його як "автомобіль" і призначити відповідну мітку [1].

Виявлення об'єктів також відоме як об'єктне виявлення або детекція об'єктів - це завдання комп'ютерного зору, яке полягає в ідентифікації та локалізації об'єктів на зображенні чи відео [1]. Це означає, що система може визначити, що на зображенні знаходиться автомобіль та намалювати прямокутник навколо нього.

Обидві архітектури, що порівнюються, є конволюційними (згортковими) нейронними мережами, які використовуються для вирішення задач комп'ютерного зору, таких як класифікація та виявлення об'єктів. Convolutional Neural Network (далі – CNN) - це клас нейронних мереж, що використовуються для роботи з великими матрицями даних, такими як зображення [2]. Згорткова нейронна мережа приз-

начена для розпізнавання об’єктів на зображеннях. CNN використовує конволюційні шари для виявлення локальних ознак у вхідних даних. Кожен шар використовує фільтри для згортки зображення та виявлення ознак, таких як границі, форми та текстури. Після цього використовуються пулінг-шари для зменшення розмірності даних та витягнення важливих ознак. Нарешті, отримані ознаки подаються на повнозв'язний шар для класифікації або виявлення об'єктів.

Архітектура InceptionResNetV2

InceptionResNetV2 є комбінацією двох попередніх архітектур - Inception та ResNet. Вона базується на концепції блоків Inception, що дозволяє ефективно використовувати різні розміри ядер для виявлення ознак на різних рівнях абстракції. Завдяки модулю ResNet архітектура використовує з'єднання залишкових з'єднань для полегшення тренування глибоких нейронних мереж і запобігання проблемі зникаючих градієнтів. На рисунку 1 наведено схему архітектури InceptionResNetV2.

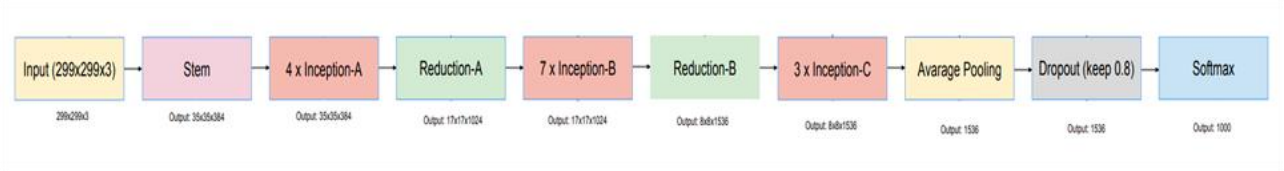

Рисунок 1 – Схема архітектури InceptionResNetV2

Основні блоки архітектури InceptionResNetV2.

Inception-ResNet-v2-A - цей блок поєднує в собі основні принципи архітектур Inception та ResNet. Він складається з кількох послідовних блоків, кожен з яких включає в себе набір змішаних (Inception) та з'єднаних залишкових (ResNet) з'єднань. Дозволяє збільшити глибину мережі та поліпшити точність виявлення ознак. Inception-ResNet-v2-B - блок включає в себе подібний до попереднього блоку набір змішаних та з'єднаних залишкових з'єднань, але з іншими параметрами та архітектурою. Він використовується для подальшого покращення глибини та точності моделі. Inception-ResNet-v2-C - цей блок має аналогічну структуру до попередніх двох, але з іншими параметрами та конфігурацією. Він також допомагає в подальшому збільшенні глибини та точності мережі.

Архітектура YOLOv5

YOLOv5 - це одна з версій алгоритму YOLO (You Only Look Once), яка відзначається високою швидкістю виявлення об'єктів та точністю. Архітектура базується на конволюційних нейронних мережах і використовує стратегію ділення зображення на сітку та одночасне прогнозування об'єктів та їхніх обмежуючих рамок. На рисунку 2 наведено схему архітектури YOLOv5.

Основні компоненти архітектури YOLOv5.

Backbone - основа архітектури відповідає за витягування ознак із вхідного зображення. Вона включає в себе набір конволюційних шарів, які послідовно аналізують зображення та виділяють важливі ознаки.

Neck - це компонент, що дозволяє зменшити просторові розміри ознак та об'єднати їх в більш високорівневі представлення. Вона використовується для підвищення точності та швидкості виявлення об'єктів.

Head відповідає за прогнозування класів об'єктів та їхніх обмежуючих рамок. Вона включає в себе декілька конволюційних шарів та вихідний шар Softmax для отримання ймовірностей класів. Використовує ознаки з neck та виконує кроки передбачення коробки та класу.

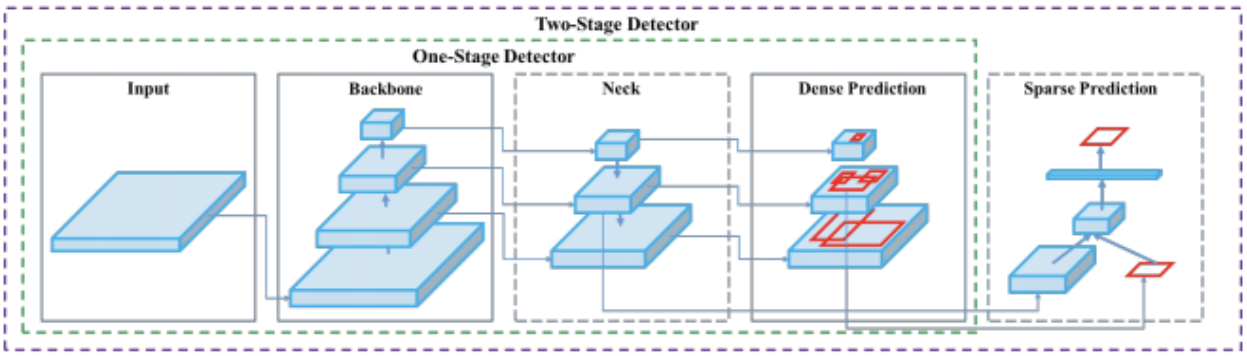

Рисунок 2 – Схема архітектури InceptionResNetV2

Результати дослідження

Для проведення дослідження було взято 300 картинок зі зображенням автомобільних номерних знаків з метою навчання моделей InceptionResNetV2 та YOLOv5 для виявлення об'єктів на зображеннях. Картинки було розділено на тренувальні – 250, та тестові – 50. Параметри навчання для обох моделей було встановлено на десять batch та сто epoch. Batch- це кількість прикладів, що обробляються моделлю перед оновленням ваг моделі. Epoch - це кількість проходів навчальним набором даних. Дані моделі будуть навчатися на CPU, тому для ефективного та швидкого навчання необхідна потужна модель процесора. Після закінчення навчання моделей проведемо тестування.

Тепер використаємо моделі у програмі. Для перевірки візьмемо п'ять картинок. На перших чотирьох картинках, моделі InceptionResNetV2 та YOLOv5 справилися без проблем, точно виявивши об'єкти на зображеннях. Однак, на п'ятій картинці модель InceptionResNetV2 не змогла знайти на зображенні номерний знак, в той час як YOLOv5 впорався з цим завданням без проблем.

Результати першої моделі, навченої на архітектурі InceptionResNetV2, показали високу точність виявлення об'єктів на зображеннях, але через брак даних для навчання не змогла визначити об’єкт на фото – рисунок 3.

Рисунок 3 – Результат п’ятого тесту з використанням InceptionResNetV2

Друга модель, навчена на архітектурі YOLOv5 показала кращі результати виявлення об'єктів на зображеннях і не допустила помилок при тестуванні. На рисунку 4 наведено приклад п’ятого тесту з використанням моделі YOLOv5.

Рисунок 4 – Результат п’ятого тесту з використанням YOLOv5

Висновки

Під час дослідження було виявлено, що для швидкого навчання моделі з малою кількістю даних краще використовувати YOLOv5, оскільки він виявився більш ефективним у цьому випадку. Модель InceptionResNetV2, хоча й показала добрі результати, але не впоралась з виявленням об'єктів на одній із тестових картинок. Тому, якщо є необхідність у швидкому навчанні або обробці зображень з малою кількістю даних, краще обрати YOLOv5. Для більшої кількості даних обидві моделі є ефективними та точними у виявленні об'єктів на зображеннях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. YOLO Algorithm: Real-Time Object Detection from A to Z [Електронний ресурс] – Режим доступу: https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z#yolo-v5
2. What are convolutional neural networks? [Електронний ресурс] – Режим доступу: https://www.ibm.com/topics/convolutional-neural-networks
3. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A., Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016. – 12p.

Бабійчук Іван Олександрович – студент групи 1СП-20Б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: youngfess4@gmail.com

Городецька Оксана Степанівна – кандидат технічних наук, доцент кафедри обчислювальної техніки Вінницького національного технічного університету, Вінниця.

Babiichuk Ivan – student of group 2SP-20b, faculty of information technologies and computer engineering, Vinnytsia National Technical University, Vinnytsia, e-mail:

Horodetska Oksana – Candidate of Technical Sciences, Assistant Professor of the Computer Techniques Chair Vinnytsia National Technical University , Vinnytsia.