DOI 10.31891/2307-5732-2024-333-2-41
УДК 004.056

ЦИВАДИЦЬ ПАВЛО
Хмельницький національний університет
https://orcid.org/0009-0004-6852-0173
e-mail: civadicpavel@gmail.com
СКРИПНИК ТЕТЯНА
Хмельницький національний університет
https://orcid.org/0000-0002-8531-5348
e-mail: tkskripnik1970@gmail.com
ВОЗНЮК ЛЕОНІД
Хмельницький національний університет
https://orcid.org/0009-0002-1152-3192

ПОРІВНЯННЯ МЕТОДІВ ВИЯВЛЕННЯ ОБ'ЄКТІВ В КОМП'ЮТЕРНОМУ ЗОРІ

Виявлення об'єктів є фундаментальним завданням комп'ютерного зору, із застосуваннями, починаючи від автономного водіння до систем спостереження. У цій статті представлено повне порівняння різних методів виявлення об'єктів. Оцінені методи включають традиційні методи, такі як логістична регресія та SVM, а також найсучасніші архітектури глибокого навчання, такі як YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), FPN (Feature Pyramid Network). , RetinaNet. YOLO надає пріоритет швидкості обробки в реальному часі, що робить його ідеальним для додатків, які потребують швидкого виявлення, наприклад, для безпілотних автомобілів. Однак такий наголос на швидкості може знизити точність порівняно з іншими методами. SSD забезпечує переконливий баланс між швидкістю та точністю, досягаючи швидшої обробки, ніж деякі методи, зберігаючи хороші можливості виявлення. FPN вирішує проблему виявлення об'єктів у різних масштабах на зображенні. Використовуючи функцію Feature Pyramid Network, він може ефективно аналізувати як малі, так і великі об'єкти в одній структурі. З іншого боку, RetinaNet зосереджується на підвищенні точності шляхом впровадження функції фокальних втрат, яка пом'якшує проблему дисбалансу класів, поширену перешкоду в задачах виявлення об'єктів, де певні класи значно перевершують інші. Для класифікації об'єктів YOLO використовує функцію втрат крос-ентропії. Ця функція вимірює різницю між прогнозованим розподілом ймовірностей класу об'єкта та фактичним розподілом класу. Мінімізація цих втрат під час навчання спрямовує модель на більш точні прогнози класу.

В роботі проведено аналіз існуючих методів виявлення об'єктів та проведено експеримент із моделлю YOLOv5, навченою на наборі даних COCO.

Ключові слова: комп'ютерний зір, виявлення об'єктів, архітектури глибокого навчання.

TSIVADITS PAVLO, SKRYPNYK TETIANA, VOZNIUK LEONID
Khmelnytskyi National University

COMPARISON OF OBJECT DETECTION METHODS IN COMPUTER VISION

Object detection is a fundamental task in computer vision, with applications ranging from autonomous driving to surveillance systems. This article presents a comprehensive comparison of various object detection methods. The methods evaluated include traditional methods such as logistic regression and SVM, as well as state-of-the-art deep learning architectures such as YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), FPN (Feature Pyramid Network), RetinaNet. YOLO prioritizes real-time processing speed, making it ideal for applications demanding swift detection, such as self-driving cars. However, this emphasis on speed might compromise accuracy when compared to other methods. SSD offers a compelling balance between speed and accuracy, achieving faster processing than some methods while maintaining good detection capabilities. FPN solves the problem of detecting objects at different scales in an image. Using the Feature Pyramid Network function, it can effectively analyze both small and large objects in the same structure. RetinaNet, on the other hand, focuses on improving accuracy by introducing a focal loss function that mitigates the class imbalance problem, a common obstacle in object detection tasks where certain classes significantly outperform others. For object classification, YOLO utilizes the cross-entropy loss function. This function measures the difference between the predicted probability distribution of an object's class and the actual class distribution. Minimizing this loss during training guides the model to make more accurate class predictions.

The paper analyzes the existing object detection methods and conducts an experiment with the YOLOv5 model trained on the COCO dataset.

Keywords: computer vision, object detection, deep learning architecture

Постановка проблеми

Виявлення об'єктів – це одне із найважливіших завдань комп'ютерного зору. Мета комп'ютерного зору – знайти та ідентифікувати об'єкти на зображенні або відео. Це завдання має широкий спектр застосувань, таких як:

- **Медична візуалізація:** діагностика захворювань за допомогою рентгенівських знімків, виявлення пухлин, декодування ланцюгів ДНК і пошуку нуклеотидів.
- **Автоматизована система розпізнавання:** розпізнавання облич, номерних знаків, дорожних знаків.
- **Транспорт:** автономні транспортні засоби.
- **Робототехніка:** навігація роботів, маніпуляція об'єктами, візуальний контроль.

Проблема полягає в тому, що існуючі методи виявлення об'єктів в комп'ютерному зорі мають різні переваги і недоліки, що ускладнює вибір оптимального методу для конкретних застосувань. Недоліки можуть включати обмежену точність виявлення, низьку швидкість обробки. Крім того, існуючі методи можуть бути витратними з точки зору обчислювальних ресурсів або складні в реалізації, тому важливо провести порівняльний аналіз різних підходів та визначити оптимальний метод для конкретного застосування в

комп'ютерному зорі.

Метою роботи є проведення аналізу існуючих методів виявлення об’єктів та експерименту. Для цього використовувалась YOLOv5 з попередньо підготовленими вагами та набір даних з 16 зображень.

Виклад основного матеріалу

YOLO – це одноетапний алгоритм розпізнавання об'єктів, який використовує згорткову нейронну мережу для прогнозування присутності, класу та обмежувальних рамок об’єктів у реальному часі. Оскільки YOLO не проходить етап пропозиції регіону та передбачає лише обмежену кількість обмежувальних рамок, він здатний дуже швидко прогнозувати наявність та розташування об’єктів на зображенні. YOLO поєднує як створення пропозицій, так і класифікацію в один крок. Це усуває потребу в окремому етапі пропозиції регіону, значно зменшуючи витрати на обчислення та забезпечуючи швидший висновок. Це значна перевага YOLO порівняно з іншими алгоритмами розпізнавання об’єктів, які використовують двоетапний підхід.

Робочий процес YOLO:

1. Навчання мережі класифікації зображень.
2. Розбиття зображення на $S \times S$ клітини. Якщо центр об’єкта потрапляє в клітину, ця клітина «відповідає» за виявлення існування цього об’єкта. Кожна клітина передбачає розташування B обмежувальних рамок, оцінку достовірності та ймовірність класу об’єкта, залежну від існування об’єкта в обмежувальній рамці [1].
   - Координати обмежувальної рамки визначаються кортежем із 4 значень – (x, y, w, h), де x і y встановлюються як зміщення розташування клітини.
   - Оцінка достовірності: $S_{conf} = Pr(obj) \times IoU$, де $Pr(obj)$ – ймовірність, IoU – перетин через об’єднання.
   - Якщо клітина містить об’єкт, вона передбачає ймовірність належності цього об’єкта до кожного класу $C_i, i = 1, \dots, K$: $\Pr(C_i \mid obj)$. Однак YOLO безпосередньо не виводить цю ймовірність. Замість цього він використовує функцію softmax як функцію активації на останньому рівні: $\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}}$, де $\vec{z}$ – вектор логітів, e^{z_i} – експоненціальна функція для вхідного вектора, $\sum_{j=1}^{K} e^{Z_i}$ – сума експонент елементів у певному каналі в межах тензора.
   - Одне зображення містить $S \times S \times B$ обмежувальних рамок, кожна рамка відповідає 4 прогнозам розташування, оцінці достовірності і K умовних ймовірностей для класифікації об'єктів.
3. Останній рівень попередньо навченої згорткової нейронної мережі модифікується для виведення тензора прогнозування розміру $S \times S \times (5B + K)$.

Функція втрат крос-ентропії YOLO використовується для оптимізації параметрів нейронної мережі під час тренування. Вона вимірює розбіжність між прогнозами моделі та істинними значеннями для даних об'єктів. Складається з двох частин: втрати локалізації для передбачення зміщення обмежувальної рамки та втрати класифікації для ймовірностей умовного класу. Параметр λ_{coord} регулює вагу втрати координат на загальну втрату . Вищі значення роблять більший акцент на точному передбаченні обмежувальної рамки, що, можливо, впливає на швидкість. Параметр λ_{noobj} регулює вагу за відсутності об’єктів на загальну втрату. Вищі значення збільшує штраф за передбачення об’єктів у порожніх клітинах сітки та робить YOLO більш зосередженим на уникненні «помилкових спрацьовувань» [2].

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + \left(\sqrt{w}_i - \sqrt{\widehat{w}_i}\right)^2 + \left(\sqrt{h}_i - \sqrt{\hat{h}_i}\right)^2] \quad (1)$$

$$L_{cls} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (\mathbb{1}_{ij}^{obj} + \lambda_{noobj}(1 - \mathbb{1}_{ij}^{obj}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \in C} \mathbb{1}_{i}^{obj} (p_i(c) - \hat{p}_i(c))^2 \quad (2)$$

$$L = L_{loc} + L_{cls}, \quad (3)$$

де $\mathbb{1}_{i}^{obj}$ – індикаторна функція про те, чи містить клітина *i* об'єкт;

$\mathbb{1}_{ij}^{obj}$ – індикаторна функція про те, чи “відповідає” обмежувальний прямокутник *j* клітини *i* за передбачення об’єкта.

YOLO має труднощі з розпізнаванням нестандартних об'єктів, а також груп дрібних об'єктів, що обумовлено використанням сітки. Крім того він демонструє чутливість до змін освітлення, має меншу стійкість до шуму та артефактів, присутніх у зображеннях, що пояснюється залежністю від згорткових нейронних мереж та навчальних даних, а також обмеженнями одноетапного підходу та функції втрат.

SSD — це алгоритм глибокого навчання, який використовується для розпізнавання об’єктів на зображеннях і відео. Представлений в 2016 році, він вперше використовув пірамідальну ієрархію ознак згорткової нейронної мережі для точного виявлення об’єктів різного розміру. Як правило, архітектура SSD зазвичай складається з базової мережі, такої як VGG або ResNet, яка попередньо навчена на великому наборі даних класифікації зображень, наприклад ImageNet. Після цієї базової мережі слідує кілька додаткових шарів, які додаються поверх базової мережі. Їх можна розглядати як пірамідальне представлення зображень у різних масштабах. Карти ознак з більшою деталізацією знаходяться на ранніх рівнях мережі та володіють кращою роздільною здатністю, а карти ознак з меншою деталізацією розміщені на пізніших рівнях мережі та мають нижчу роздільну здатність. На відміну від YOLO, SSD не розбиває зображення на сітки довільного розміру, а передбачає зміщення попередньо визначених прив'язок кожного місця на карті ознак. Кожний блок має

фіксований розмір і положення відносно відповідної клітини. Усі блоки прив'язки розміщують всю карту об'єктів у згортковий спосіб.

FPN – архітектура, яка розроблена для поєднання ознак з різних рівнів згорткової мережі, щоб краще виявляти об'єкти в різних масштабах. Мережа бере зображення як вхідний сигнал і надає на виході пропорційно масштабовані карти ознак на кількох рівнях. Ці карти ознак містять інформацію про зображення на різних рівнях деталізації. Побудова піраміди включає шляхи знизу вгору та шляхи зверху вниз. У висхідному шляху магістральна мережа, як ResNet, використовується для вилучення ознак із зменшенням рівня просторової роздільної здатності. У міру того, як рівні роздільної здатності зменшуються, семантичне значення карт ознак збільшується, як вказує товщина рамок синього кольору. У низхідному шляху карти ознак об'єднуються, щоб мати багате семантичне значення та точну просторову інформацію. Завдяки потужності побудови карт ознак із багатим семантичним значенням і високою просторовою точністю, FPN широко використовується в багатьох проблемах комп'ютерного зору:

RetinaNet – одноетапна модель виявлення об'єктів, яка добре працює з щільними та малими об'єктами. RetinaNet використовує FPN для отримання багаторівневих ознак. Нижчі рівні FPN мають високу роздільну здатність, що дозволяє краще локалізувати об'єкти, а верхні рівні мають більш абстрактні ознаки, що допомагає класифікувати їх. На кожному рівні FPN RetinaNet застосовує дві підмережі:

- Підмережа класифікації: Передбачає ймовірність присутності об'єкта в кожному просторовому місці для кожного блоку прив'язки та класу об'єктів.
- Підмережа регресії: Виконує регресію (прогнозування) координат обмежувальних рамок, які точно визначають положення об'єктів на зображенні.

У RetinaNet, як і в більшості алгоритмів виявлення об'єктів, існує проблема дисбалансу між фоном, який не містить об'єктів, і переднім планом, який містить об'єкти інтересу.Фокальна втрата призначена для присвоєння більшої ваги складним прикладам (фон з частковим об'єктом або з шумною текстурою), які легко класифікувати неправильно, а також зменшення ваги простих прикладів (порожній фон). Ця функція втрат додає ваговий коефіцієнт $(1-p_t)^\gamma$ до кожного члена в перехресній ентропії, так що вага є малою, коли p велике , і тому легкі приклади мають меншу вагу [6].

$$FL(p_t) = -(1-p_t)^\gamma \, log\, p_t \tag{4}$$

Для кращого контролю форми вагової функції RetinaNet використовує α – збалансований варіант функції втрат, який працює найкраще, коли $\alpha = 0.25, \gamma = 2$ або $\alpha = 0.75, \gamma = 5$.

$$FL(p_t) = -\alpha(1-p_t)^\gamma \, log\, p_t \tag{5}$$

Оптимальне значення α і γ може варіюватися залежно від набору даних, архітектури мережі та завдання. Низькі значення α роблять функцію втрат схожою на перехресну ентропію, в той час як високі значення надають більшу вагу фоновим пікселям, що може допомогти при сильному дисбалансі класів. Тобто коли $\alpha = 0$, функція не робить жодної різниці між фоновими та об'єктними пікселями. Параметр γ впливає на ступінь штрафу за неправильне прогнозування складних зразків. Аналогічним шляхом з α значення γ роблять функцію втрат більш або менш жорсткою для складних зразків.

Ці графіки показують, як фокальна втрата змінюється в залежності від ймовірності правильної класифікації прикладу для чотирьох значень γ та α.

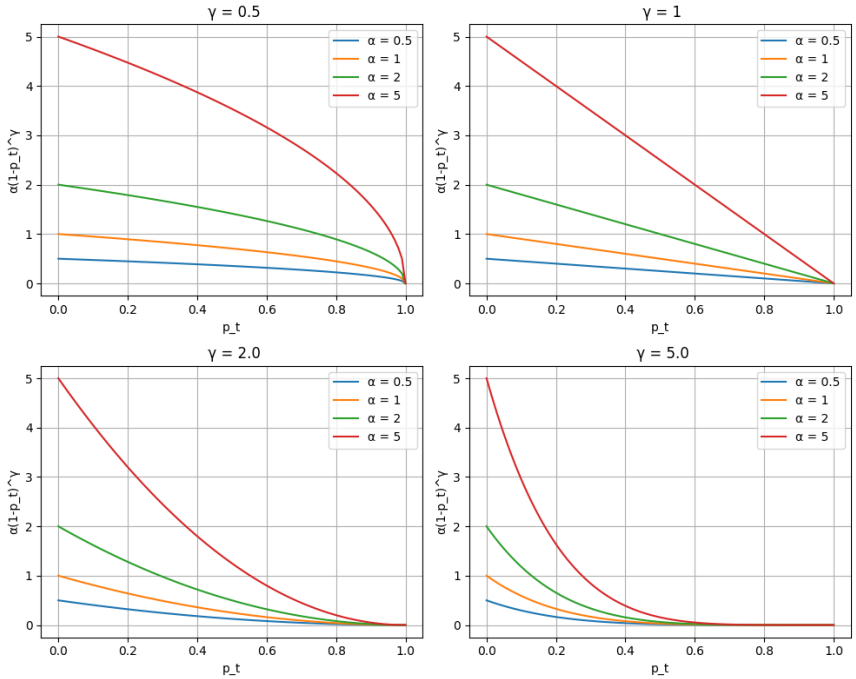

Рис. 1. Вплив параметрів γ та α на фокальну втрату

Результати експерименту

Модель YOLOv5 була навчена з використанням пакету з 16 зображень і розміром вхідного зображення 640x640 пікселів протягом 3 ітерацій. Набір даних COCO128 використовувався для навчання, а попередньо підготовлені ваги YOLOv5 використовувалися як початкова точка. Побудовані криві впевненості і точності-відкликання.

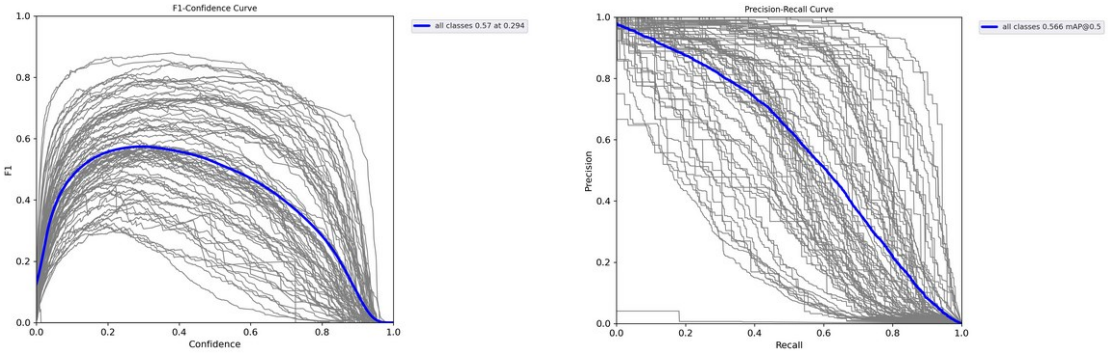

Рис. 2.

Висновки

Таким чином, проведено аналіз існуючих методів виявлення об’єктів. Досліджено можливості використання моделі YOLOv5 для виявлення об'єктів. Навчання показало, що YOLOv5 можна ефективно використовувати для виявлення об’єктів, що демонструє баланс між точністю та швидкістю.

Необхідно зазначити, що навчання має ряд обмежень, таких як невеликий розмір набору даних та короткий час навчання.

References

1. Object Detection With Deep Learning: A Review / Z.-Q. Zhao та ін. *IEEE Transactions on Neural Networks and Learning Systems*. 2019. Т. 30, № 11. С. 3212–3232. https://doi.org/10.1109/tnnls.2018.2876865.

2. Object Detection in 20 Years: A Survey / Z. Zou та ін. *Proceedings of the IEEE*. 2023. С. 1–20. https://doi.org/10.1109/jproc.2023.3238524.

3. Object Detection: SSD Vs. YOLO | Baeldung on Computer Science. *Baeldung on Computer Science*. Available at: https://www.baeldung.com/cs/object-detection-ssd-yolo

4. Papers with Code - Darknet-53 Explained. *The latest in Machine Learning | Papers With Code*. Available at: https://paperswithcode.com/method/darknet-53

5. Papers with Code - SSD Explained. *The latest in Machine Learning | Papers With Code*. Available at: https://paperswithcode.com/method/ssd

6. Potrimba P. What is R-CNN?. *Roboflow Blog*. Available at: https://blog.roboflow.com/what-is-r-cnn/

7. R-CNN | Region Based CNNs - GeeksforGeeks. *GeeksforGeeks*. Available at:

8. https://www.geeksforgeeks.org/r-cnn-region-based-cnns/

9. Papers with Code - FPN Explained. *The latest in Machine Learning | Papers With Code*. Available at: https://paperswithcode.com/method/fpn

10. Object Detection Part 4: Fast Detection Models. *Lil'Log*. Available at: https://lilianweng.github.io/posts/2018-12-27-object-recognition-part-4/#image-pyramid