

ОБЗОР ВОЗМОЖНОСТЕЙ YOLO V8 В РАМКАХ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Бурень М.Н.

Белорусский государственный университет информатики и радиоэлектроники¹
г. Минск, Республика Беларусь

Татур М.М. – доктор тех. наук, профессор каф. ЭВМ

Аннотация. В данной работе проведен анализ функциональных возможностей модели YOLO v8 при решении задачи классификации объектов на изображении. Разобраны режимы работы и виды подмоделей, а также соотношение их эффективности и производительности.

Ключевые слова. YOLO, сеть обнаружения объектов, набор данных, задача прогнозирования, классификация.

Введение. Классификация объектов на изображении одна из основных задач машинного зрения. Несмотря на то, что это тривиальная задача для человека, ее автоматизация долгое время не представлялась возможной. Причиной этому служило отсутствие простого алгоритма, по которому было бы возможно классифицировать большое количество объектов на изображении. Задача могла быть решена с использованием нейронных сетей, но долгое время вычислительных мощностей компьютеров для этого было недостаточно. Ситуация изменилась в XXI веке [1] с развитием многопроцессорных (многоядерных систем) и графических ускорителей, которые позволили распараллелить вычисления, используемые в работе и обучении нейронных сетей.

История и Архитектура моделей YOLO. Оригинальная версия YOLO (You Only Look Once) была создана Джозефом Редмоном в 2016 году [2]. Исходная модель YOLO была первой сетью обнаружения объектов, которая объединила проблему формирования ограничивающих рамок и идентификации классов в одной сквозной дифференцируемой сети. Некоторые модели обнаружения объектов рассматривают распознавание как задачу, состоящую из двух частей. Сначала определяется области интереса (ограничивающая рамка), в которой находится объект. Во-вторых, классифицируют эту конкретную область интересов. Рассматривая задачу обнаружения как однократный регрессионный подход для определения ограничивающих рамок, модели YOLO часто оказываются более производительными чем их аналоги, что упрощает их использование на устройствах с ограниченными вычислительными возможностями.

В данной статье будет рассматриваться одна из актуальных версий YOLO v8. Алгоритм ее работы приведена на рисунке 1

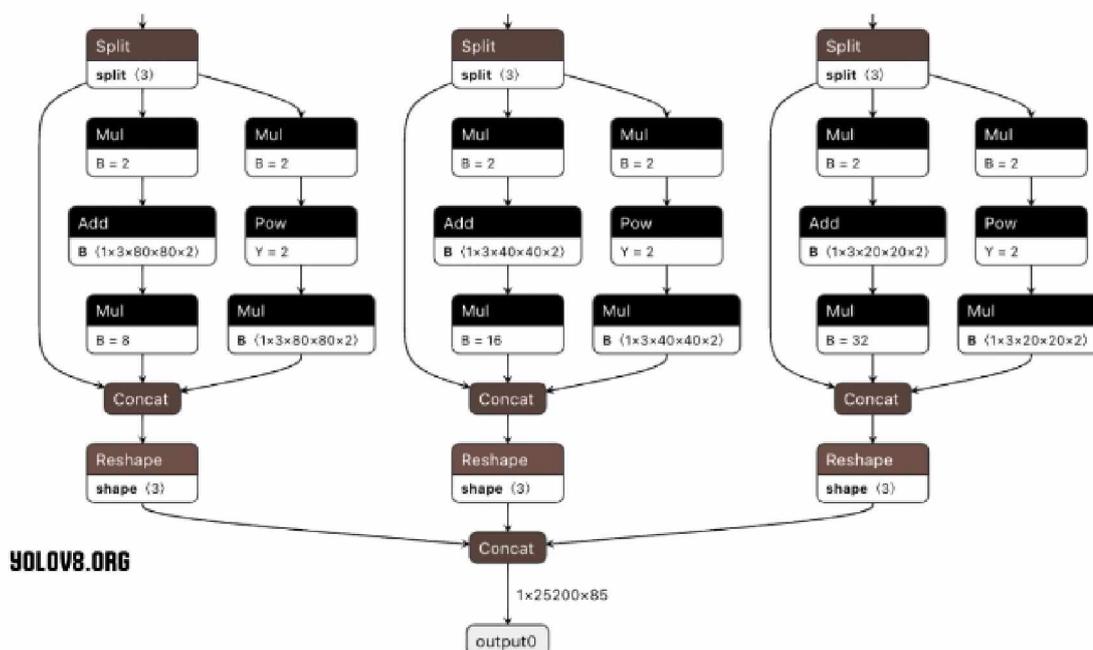


Рисунок 1 – Алгоритм работы модели YOLO v8 [3]

Функциональные возможности. В данном разделе будут разобраны основные функциональные возможности модели YOLO v8. Работа с моделью проводилась на следующем оборудовании:

- Центральный процессор Intel core i5 10400f.
- Оперативная память 16GB gddr 4 (2400Gh).
- Графический ускоритель Nvideo Geforce RTX 2060 Super.

Сама модель и ее предварительно обученные на наборе данных COCO [4] веса, используемые далее, были загружены с официального сайта [5].

Модель работает на базе языка программирования Python3. Возможно использование как через интерфейс командной строки (CLI), так и через программный интерфейс python (API). Входными данными для системы могут являться отдельные изображения, видео файлы, а также видео поток. Выходными данными являются информация о обнаруженных на изображениях объектах (их класс, расположение и другие данные в зависимости от режима работы и типа модели). Данная информация может быть автоматически наложена на исходное изображение и сохранена/выведена в отдельный файл. YOLO v8 может использовать как CPU, так и GPU. Для работы с последним необходимо установить дополнительное программное обеспечение (CUDA, OPENGL и т.д.). Разница в производительности составила 4.68 раза при работе с видеопотоком на тестовой системе.

Основные рабочие режимы:

1. Прогнозирование (predict) – определяет объекты на изображение (см. рисунок 2а).
2. Сопровождение (track) – определяет объекты с присваиванием объекту идентификатора и сохранением его на протяжении сцены (см. рисунок 2б).



Рисунок 2 — Основные рабочие режимы: а — прогнозирование; б — сопровождения

Как видно из результатов режим сопровождения при прочих равных (одно и то же изображение и одинаковые веса) определяет меньше объектов на изображении, в тоже время на тестовом стенде с использованием GPU производительность задачи прогнозирования оказалась в 2.72 раза выше (разница в затраченном на обработку времени).

Подвиды моделей:

1. Определение (detect) – определяет объекты на изображение. При этом определяется только прямоугольная область, в рамках которой находится объект (см. рисунок 3а).
2. Сегментации (segment) - определяет объекты на изображение и точные границы определенного объекта (см. рисунок 3б).

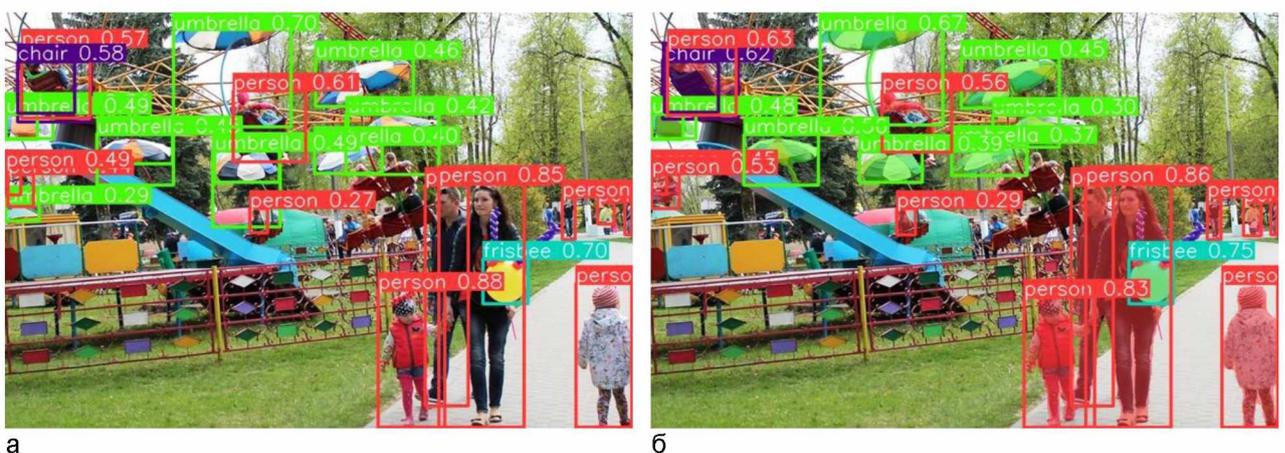


Рисунок 3 — Основные виды моделей: а — определения; б — сегментации

3. Классификация (Classification) – определяет класс (что изображено) изображения. Фактически рассчитывает несколько наиболее вероятных классов (см. рисунок 4а и рисунок 4б).



а



б

Рисунок 4 — Модель классификации: а — ошибочное определение; б — правильное определение

Ошибка в определении класса объекта на изображении объясняется отсутствием в обучающем наборе “COCO” объектов данного типа.

4. Определение позы (Pose Estimation) - определяет объекты на изображении и расположение ключевых точек идентифицированного объекта (см. рисунок 5).



Рисунок 5 - Определение позы

5. Определение прямоугольных объектов (Oriented Bounding Boxes Object Detection) – определяет объекты на земле при аэрофотосъемке (см. рисунок 6).



Рисунок 6 - Определение прямоугольных объектов

Базовые веса моделей, представленные на официальном сайте, дополнительно разделены на 5 классов согласно размеру. Вес модели на прямую влияет на производительность. Например, самая легкая модель, предназначенная для определения объектов (YOLOv8n) в режиме прогнозирования, обрабатывала тестовый видео файл 31.67 секунды. Самая тяжелая модель аналогичного назначения (YOLOv8x) такую же операцию выполняла 79.01 секунды, что в 2,49 раза медленнее. Результаты для всех моделей, а также их вес приведены в таблице 1:

Таблица 1 – Зависимость скорости выполнения от веса модели.

Параметр	YOLOv8n	YOLOv8s	YOLOv8m	YOLOv8l	YOLOv8x
Вес (МБ)	6.23	21.5	49.7	83.7	130
Время выполнения (с)	31.67	32.19	40.32	57.23	79.01

Как видно из таблицы рост размера модели не ведет к прямо пропорциональному росту сложности вычислений, в тоже время точность определения объектов растет. На рисунках 7а и 7б представлена разница в определении объектов самой легкой (YOLOv8n) и самой тяжелой (YOLOv8x) моделью для распознавания. Из примера видно, что более тяжелая модель одновременно распознала больше объектов и более точно (с точки зрения вероятности) определила те, что смогла определить легкая.



а

б

Рисунок 7 — Результаты работы моделей: а — YOLOv8n; б — YOLOv8x

Обучение, а также валидация являются вспомогательными режимами работы системы. Для обучения необходимо выбрать тип модели и разметить данные в соответствии со сделанным выбором. Разметка представляет собой выделение классов объектов на изображениях (в случае базовой модели определения достаточно выделить прямоугольные границы объекта). Далее размеченные изображения нужно разместить определенным образом в отдельную папку согласно документации [5] и запустить обучение. Имеется ряд настраиваемых параметров, влияющих на время и качество обучения:

1. Количество итераций.

2. Предельное время обучения (переопределяет предыдущий параметр если задано).
3. Размер изображения.
4. Кол-во изображений на итерации.

При необходимости обучение можно проводить за несколько этапов. Возобновление обучения из ранее сохраненного состояния — важная функция при работе с моделями глубокого обучения. Это может пригодиться в различных сценариях, например, когда процесс обучения был неожиданно прерван или когда вы хотите продолжить обучение модели с новыми данными или в течение большего количества эпох.

Когда обучение возобновляется, YOLO загружает веса из последней сохраненной модели, а также восстанавливает состояние оптимизатора, планировщик скорости обучения и номер эпохи. Это позволяет продолжить тренировочный процесс с того места, где он был остановлен.

Вывод. Модель YOLO v8 позволяет решать задачу распознавания изображений на разнообразном оборудовании (в зависимости от производительности можно подобрать необходимую размерность модели). Помимо базового распознавания, возможно также сопровождение объектов в видео потоке, определение точных границ заданного объекта либо его контрольных точек. Имеющиеся базовые веса позволяют идентифицировать 80 категорий объектов [4], в случае если этого недостаточно, модель позволяет провести дообучение.

Список использованных источников:

1. Лекун Ян. Как учится машина: Революция в области нейронных сетей и глубокого обучения. Альпина PRO, 2021. – 335 с.
2. What is YOLO? The Ultimate Guide. [Электронный ресурс] / Roboflow - Режим доступа: <https://blog.roboflow.com/guide-to-yolo-models> – Дата доступа: 06.04.2024.
3. YOLOv8 Architecture: A Deep Dive into its Architecture. [Электронный ресурс] / Yolov8.org- Режим доступа: <https://yolov8.org/yolov8-architecture/> – Дата доступа: 07.04.2024.
4. COCO Dataset. [Электронный ресурс] / COCO - Режим доступа: <https://cocodataset.org/#home> – Дата доступа: 05.04.2024.
5. Ultralytics YOLOv8 Docs. [Электронный ресурс] / Ultralytics - Режим доступа: <https://docs.ultralytics.com/> – Дата доступа: 05.04.2024.

OVERVIEW OF THE CAPABILITIES OF YOLO V8 WITHIN THE FRAMEWORK OF SOLVING THE PROBLEM OF IMAGE RECOGNITION

Buren M.N.

Belarusian State University of Informatics and Radioelectronics¹, Minsk, Republic of Belarus

Tatur M.M. – Dr. of Sci. (Eng.), Professor at the Computer department

Annotation. In this work, we analyzed the functionality of the YOLO v8 model when solving the problem of classifying objects in an image. The operating modes and types of submodels, as well as the relationship between their efficiency and productivity, are analyzed.

Keywords. YOLO, object detection network, dataset, prediction task, classification.