Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ7 «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

«Классификация известных методов подсчета количества человек на видео»

Студент **ИУ7И-72Б** _____ **М. Х. Фам**

(Подпись, дата) (И.О.Фамилия)

Руководитель _____ **Т. А. Никульшина**

(Подпись, дата) (И.О.Фамилия)

Рекомендованная руководителем НИР оценка: _____

2024 г.

РЕФЕРАТ

Расчетно-пояснительная записка 22 с., 7 рис., 1 табл., 5 источн., 1 прил. YOLO, SSD, FAST R-CNN, MASK R-CNN.

Цель работы — сравнение различных методов и алгоритмов, используемых для подсчета количества людей на видео.

В результате проведенной работы был осуществлен анализ предметной области, касающейся методов подсчета количества людей на видео, в ходе которого были выделены ключевые критерии для сравнения рассматриваемых методов, а также проведена их детальная классификация, что позволило систематизировать существующие подходы и выявить их сильные и слабые стороны в контексте различных сценариев применения.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие сокращения и обозначения.

YOLO — You Only Look Once

SSD — Single Shot Detector

CNN — Convolutional neural network

FPS — Frames Per Second

ВВЕДЕНИЕ

Технология обнаружения объектов существует вообще везде. Наиболее очевидным приложением является программное обеспечение для разблокировки по распознаванию лиц на телефонах или системы камер видеонаблюдения в магазинах и складах.

С каждым годом её применение становится всё более актуальным, от обеспечения безопасности в общественных местах до автоматизации процессов в производстве [1]. Эта технология позволяет распознавать и отслеживать объекты в реальном времени, что открывает новые возможности для инноваций и улучшения качества жизни. В условиях стремительного развития технологий и увеличения объёмов данных, обнаружение объектов становится неотъемлемой частью многих систем, включая автономные транспортные средства, системы видеонаблюдения, медицинские приложения и многое другое. Развитие этой технологии не только повышает уровень безопасности и эффективности, но и способствует созданию более умных и адаптивных решений для решения актуальных задач современности.

Целью работы является классификация методов подсчета количества человек на видео. Для достижения поставленной цели необходимо решить следующие задачи:

1) провести анализ предметной области методов подсчета количества человек на видео;

2) провести обзор существующих методов подсчета количества на видео;

3) выделить критерии сравнения рассматриваемых методов и на этом основе классифицировать эти методы.

1 Анализ предметной области

1.1 Задача подсчета количества человек на видео

Подсчет количества человек на видео — важная задача компьютерного зрения, которая используется в системах видеонаблюдения, анализа поведения, управлении потоками людей и других приложениях. Она предполагает автоматическое определение количества людей в кадре на основе анализа изображений или видеопотока.

Основные этапы:

1) Обнаружение объекта — определение местоположения людей на каждом кадре видео. Современные методы, такие как YOLO, Faster R-CNN или SSD, позволяют определять местоположение объектов на каждом кадре с высокой точностью и скоростью.

2) Классификация объектов — подтверждение, что обнаруженные объекты действительно являются людьми. Это достигается с помощью методов классификации, основанных на глубоких нейронных сетях, которые обучены различать людей от других объектов.

Сложности задачи подсчета связаны с факторами, такими как перекрытие людей, изменения условий освещения и динамика сцены. Тем не менее, развитие глубокого обучения и методов компьютерного зрения позволяет успешно справляться с этими вызовами. Развитие современных методов детекции и глубокого обучения позволяет решать её с высокой точностью, что открывает широкие перспективы для применения в реальном времени.

Этот процесс представлен на рисунке 1.1.

Рисунок 1.1 – Пример подсчета количества человек на видео

1.2 Существующие методы подсчета количества человек на видео

1.2.1 Алгоритм YOLO

YOLO (You Only Look Once) — это один из самых популярных алгоритмов для обнаружения объектов в реальном времени. Он был разработан для решения задачи классификации объектов на изображениях с высокой скоростью и точностью. YOLO продемонстрировал свою способность быстро и эффективно идентифицировать, находить и распознавать объекты на изображениях. Вместо того, чтобы обрабатывать изображение как одно изображение, YOLO делит изображение на сетку из небольших ячеек. Каждая ячейка сетки считается отдельным «частичным изображением» и может принадлежать другому классу объектов. Эта идея позволяет YOLO обрабатывать несколько объектов на изображении за одно сканирование.

Такой подход помогает YOLO добиться впечатляющей производительности и высокой точности. Он решает такие проблемы, как перекрывающиеся объекты, частично затемненные объекты, объекты, появляющиеся на изображении в разных положениях и размерах, а также наличие фона изображения.

YOLO использует поля привязки для измерения и прогнозирования положения и взаимного расположения объектов, что позволяет учитывать множество различных форм объектов. В то же время YOLO также использует технику немаксимального подавления, чтобы исключить ненужные результаты и сохранить только самые важные объекты.

YOLOv1 — первоначальная модель YOLO рассматривала обнаружение объектов как проблему регрессии, что было значительным сдвигом от традиционного подхода к классификации.

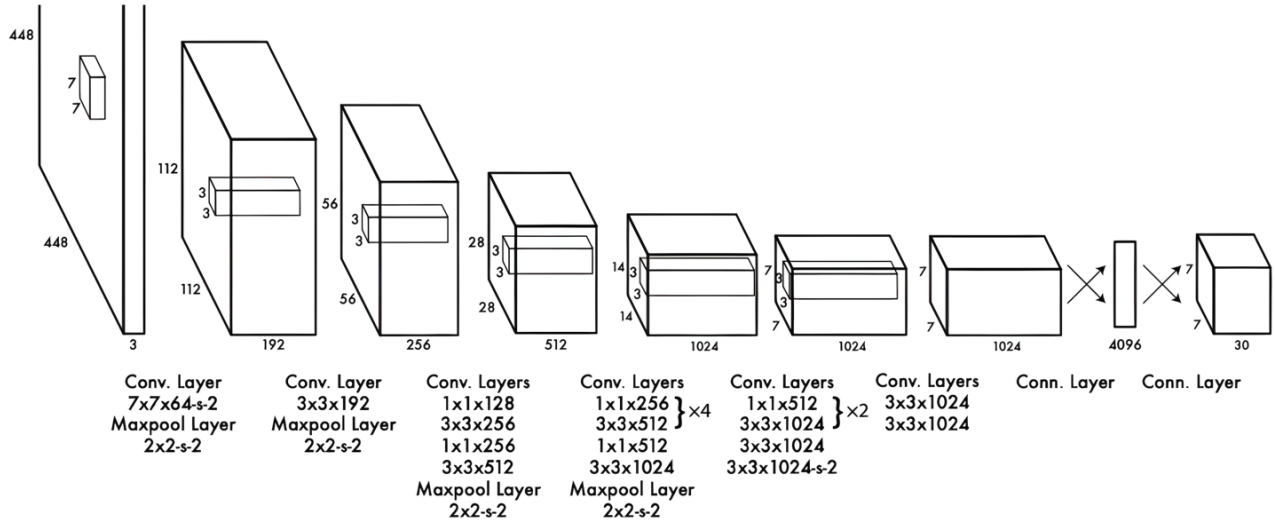

Рисунок 1.2 – Архитектура алгоритм YOLOv1

В ходе тестирования YOLOv1 происходит умножение условных вероятностей классов на прогнозируемые значения уверенности для отдельных ячеек. Это позволяет получить оценки для каждого блока, которые соответствуют конкретному классу, согласно формуле 1.1

$$Pr(Class_i|Object) \times Pr(Object) \times IOU_{pred}^{truth} = Pr(Class_i) \times IOU_{pred}^{truth} \quad (1.1)$$

Она использовала одну сверточную нейронную сеть (CNN) для обнаружения объектов на изображениях, разделяя изображение на сетку, делая несколько прогнозов на ячейку сетки, отфильтровывая прогнозы с низкой достоверностью, а затем удаляя перекрывающиеся блоки для получения окончательного

вывода.

YOLOv5 представлен в четырех основных версиях: маленькая (s), средняя (m), большая (l) и очень большая (x), каждая из которых обеспечивает постепенно возрастающий уровень точности. При этом каждая из версий требует разного времени на обучение: чем больше и точнее модель, тем дольше процесс тренировки.

На рисунке 1.3 представлено сравнение разных версий YOLOv5.

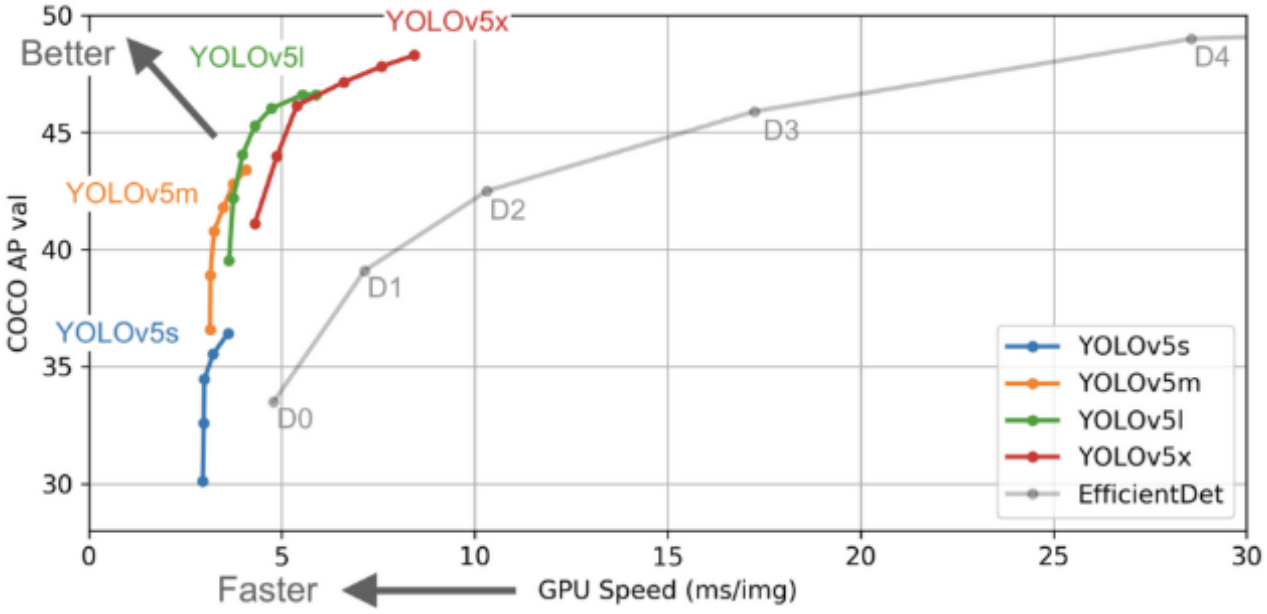

Рисунок 1.3 – Сравнение разных версий YOLOv5

Изображение проходит через входной слой (input) и передается в основную сеть (backbone) для извлечения признаков. Основная сеть получает карты признаков различных размеров, которые затем объединяются через сеть слияния признаков (neck), чтобы в итоге сформировать три карты признаков: P3, P4 и P5 (в YOLOv5 размеры выражаются как 80×80, 40×40 и 20×20). Эти карты используются для обнаружения соответственно мелких, средних и крупных объектов на изображении.

После того как три карты признаков передаются в прогнозирующую голову (head), выполняются расчёт уровня уверенности (confidence) и регрессия ограничивающих рамок (bounding-box regression) для каждого пикселя карты признаков, используя заранее заданные якоря (prior anchors). Таким образом, получается многомерный массив (BBoxes), содержащий информацию о классе объекта, уровне уверенности класса, координатах рамки, ширине и высоте.

Затем, задавая соответствующие пороговые значения (confthreshold,

objthreshold), фильтруется ненужная информация из массива. После этого выполняется процесс подавления немаксимумов (non-maximum suppression, NMS), чтобы получить финальную информацию об обнаруженных объектах.

Архитектура алгоритма YOLOv5 представлена на рисунке 1.4

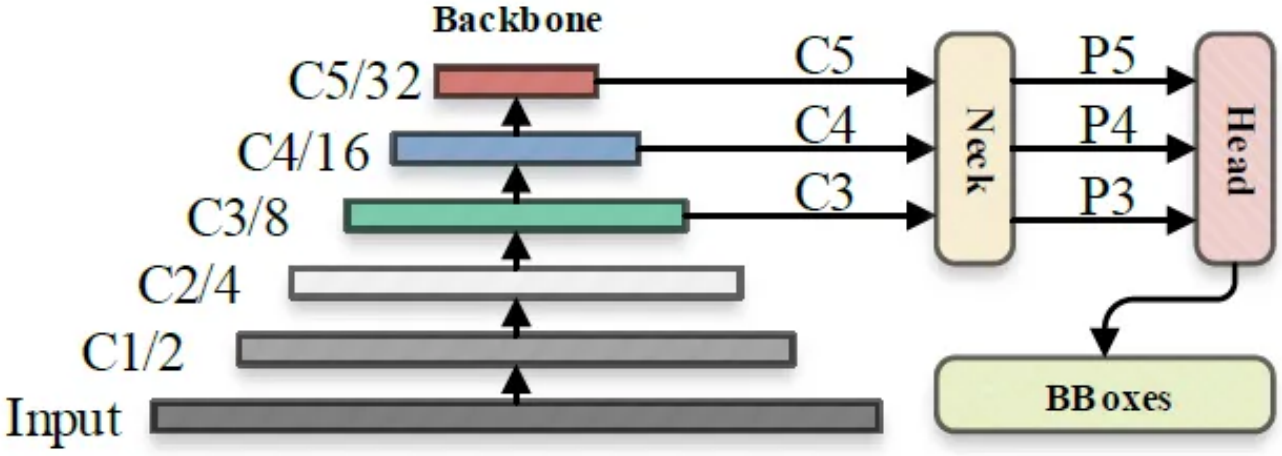

Рисунок 1.4 – Архитектура YOLOv5

1.2.2 Алгоритм SSD

SSD (Single Shot Detector) — это алгоритм для обнаружения объектов, который также ориентирован на высокую скорость и точность. Он был разработан для решения задачи локализации и классификации объектов на изображениях в реальном времени [2]. Основная идея SSD заключается в том, что он использует однопроходную нейронную сеть для предсказания ограничивающих рамок и классов объектов, что позволяет ему эффективно обрабатывать изображения.

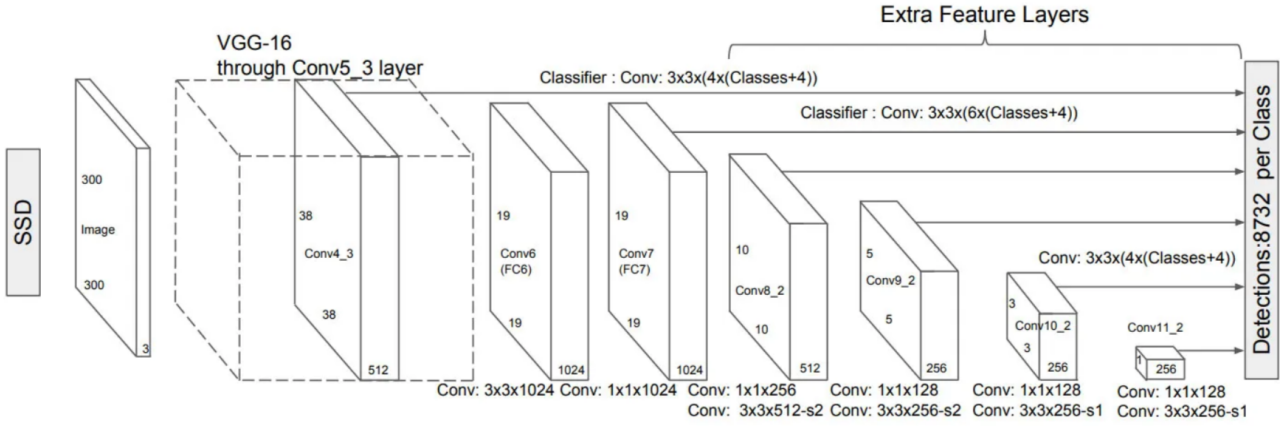

Рисунок 1.5 – Архитектура алгоритм SSD [2]

Основные характеристики алгоритма SSD можно описать следующими приведенными факторами:

1) Однопроходная архитектура: Как и YOLO, SSD выполняет обнаружение объектов за один проход через сеть, что значительно ускоряет процесс по сравнению с традиционными методами, требующими многократного анализа изображения.

2) Многоуровневая структура: SSD использует несколько уровней (или слоев) для предсказания объектов, что позволяет ему обнаруживать объекты разных размеров. На каждом уровне сети генерируются ограничивающие рамки и вероятности классов, что улучшает точность обнаружения.

3) Использование различных аспектов: Алгоритм SSD применяет различные соотношения сторон и масштабы для ограничивающих рамок, что позволяет ему более эффективно обнаруживать объекты с различными формами и размерами.

4) Быстрая обработка: SSD обеспечивает высокую скорость обработки, что делает его подходящим для приложений, требующих реального времени, таких как видеонаблюдение, автономные транспортные средства и мобильные устройства.

5) Обучение на больших наборах данных: SSD обучается на больших аннотированных наборах данных, что позволяет ему эффективно распознавать различные классы объектов и адаптироваться к различным условиям.

1.2.3 Модифицированные алгоритмы R-CNN

R-CNN (Regions with Convolutional Neural Networks) представляет собой значительный шаг вперед в области обнаружения объектов, который был предложен Россом Бахи и его коллегами в 2014 году. Алгоритм использует двухступенчатый подход, который сочетает в себе методы селекции регионов и глубокое обучение. На первом этапе R-CNN генерирует набор предложений о регионах интереса (region proposals) с помощью алгоритма Selective Search, который выделяет потенциальные области, содержащие объекты. Эти регионы

затем обрабатываются с использованием сверточной нейронной сети (CNN), что позволяет извлекать высокоуровневые признаки из каждого региона.

На втором этапе R-CNN применяет классификатор, обученный на извлеченных признаках, для определения класса объекта в каждом предложенном регионе. Для повышения точности алгоритм использует метод SVM (Support Vector Machine) для классификации и регрессию для уточнения границ объектов. Несмотря на свою эффективность, R-CNN имеет некоторые недостатки, такие как высокая вычислительная сложность и длительное время обработки, что связано с необходимостью обработки каждого региона отдельно. Эти ограничения стали основой для разработки более совершенных моделей, таких как Fast R-CNN и Faster R-CNN, которые стремятся улучшить скорость и точность обнаружения объектов, сохраняя при этом преимущества, заложенные в оригинальной архитектуре R-CNN.

Алгоритм Fast R-CNN

Fast R-CNN — улучшенная версия R-CNN, разработанная Россом Гиршиком, которая устраняет основные недостатки R-CNN, такие как высокая вычислительная сложность и длительное время обработки [3]. Fast R-CNN использует слой ROI Pooling для преобразования предложений регионов (ROIs) в векторы фиксированной длины, что позволяет выполнять свёрточные операции один раз для всего изображения и делить результаты между всеми регионами. Это значительно ускоряет обработку и уменьшает затраты памяти.

Вместо трёх отдельных этапов (генерация регионов, извлечение признаков и классификация) Fast R-CNN объединяет их в единую архитектуру. В результате модель работает быстрее и эффективнее. Однако Fast R-CNN всё ещё зависит от медленного алгоритма Selective Search для генерации предложений регионов, что ограничивает её производительность. Этот недостаток исправляется в Faster R-CNN, где вводится собственная сеть для генерации регионов.

Архитектура алгоритма Fast R-CNN представлена на рисунке 1.6

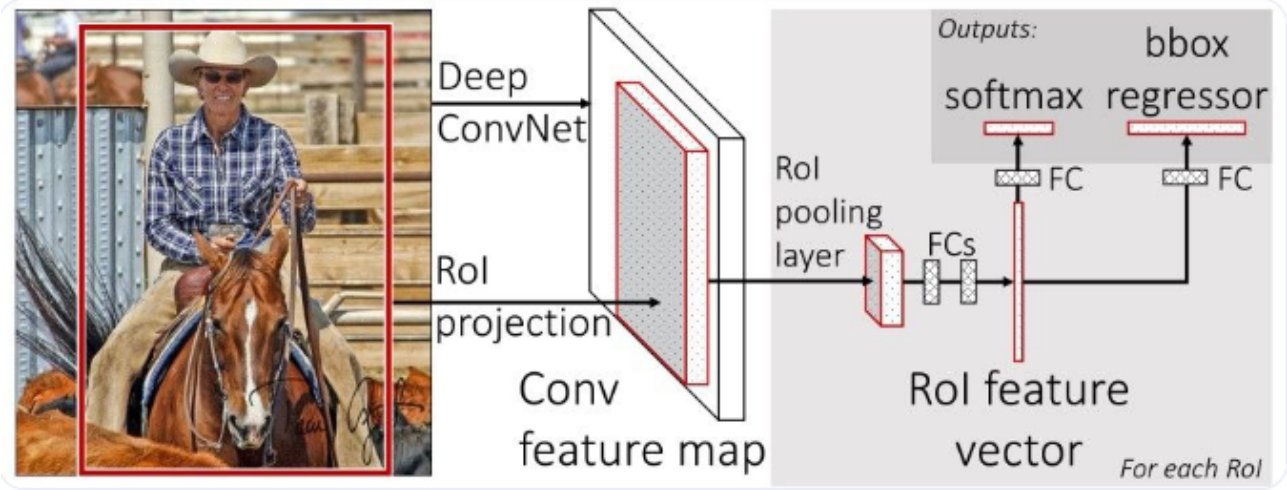

Рисунок 1.6 – Архитектура Fast R-CNN [3]

Алгоритм Mask R-CNN

Mask R-CNN — это нейронная сеть, предназначенная для задачи сегментации объектов (instance segmentation) в компьютерном зрении [4]. Она может выделять отдельные объекты на изображении или видео, предоставляя ограничивающие рамки (bounding boxes), классы объектов и их маски на уровне пикселей.

Архитектура алгоритма Mask R-CNN представлена на рисунке 1.7

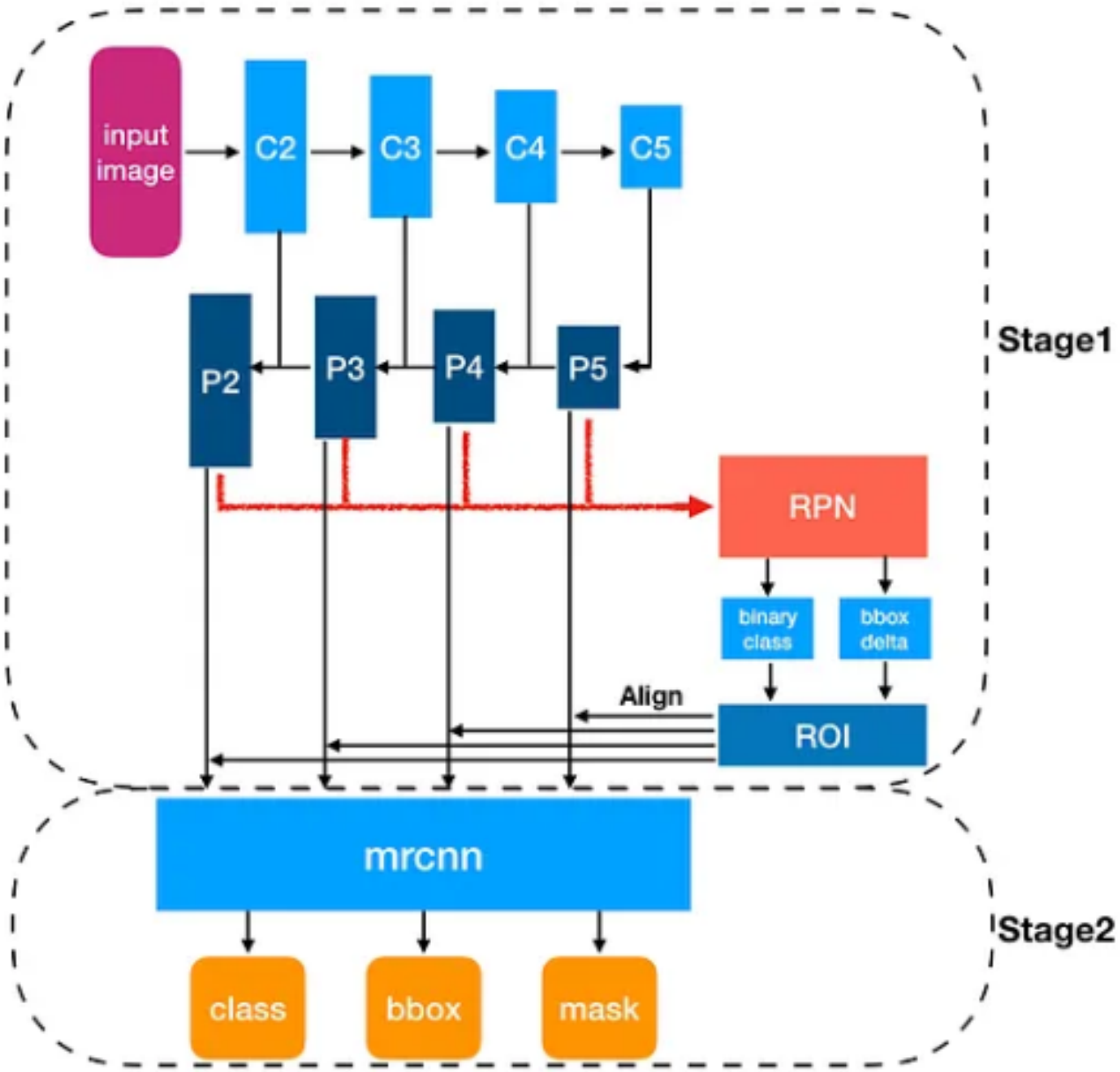

Рисунок 1.7 – Архитектура Mask R-CNN

Архитектура Mask R-CNN состоит из двух этапов. На первом этапе сеть генерирует предложения регионов (Region Proposal Network, RPN), которые могут содержать объекты. Для этого используется Feature Pyramid Network (FPN), которая обеспечивает извлечение признаков на разных масштабах. FPN включает в себя сверточную сеть (например, ResNet или VGG) для анализа изображения и механизм, сохраняющий информацию на разных уровнях разрешения.

На втором этапе сеть уточняет полученные регионы, определяет классы объектов, уточняет координаты ограничивающих рамок и создает маски объектов. В отличие от первого этапа, на этом этапе используется метод

ROIAlign, который позволяет точно сопоставить области на карте признаков с исходным изображением, обеспечивая высокую точность результатов [4].

Главное преимущество Mask R-CNN — способность работать с объектами на разных масштабах благодаря архитектуре FPN и использованию якорей (anchors). Это позволяет сохранять пространственные отношения между объектами на исходном изображении и их признаками на карте. Mask R-CNN активно используется в задачах медицинской диагностики, автономного вождения и видеоаналитики.

2 Сравнение методов подсчета количества человек на видео

2.1 Критерии сравнения методов подсчета количества человек на видео

Для сравнения рассматриваемых методов будут использоваться следующие критерии [5]:

1) Точность (mAP).

2) Скорость (FPS).

3) Поддержка сегментации.

Критерий «mAP» определяет, насколько хорошо метод распознает людей, избегая как ложных срабатываний, так и пропущенных объектов. Критерий «mAP» вычисляется по следующей формуле 2.1.

$$mAP = 1/N \sum_{i=1}^{N} AP_i \tag{2.1}$$

, где N — количество классов, AP (average precision) — метрика, используемая для оценки точности работы детекторов объектов.

Критерий «Скорость» метода подсчета людей на видео определяется количеством кадров, обрабатываемых за секунду (FPS). Высокий FPS (30+) важен для задач реального времени, например, видеонаблюдения, а низкий FPS (<10) подходит для аналитики, где скорость менее критична. FPS является ключевым показателем эффективности метода.

Критерий «Поддержка сегментации» определяет возможность выделять маски объектов на уровне пикселей.

2.2 Результаты сравнения

В таблице 2.1 представлено сравнение рассматриваемых методов [5].

Таблица 2.1 – Сравнение методов обнажурения объектов на основе сформированных критериев

Метод	mAP	FPS	Поддержка сегментации
YOLOv1	63	50	Нет
YOLOv5	68-73	60-120	Нет
SSD	66	35-45	Нет
Fast R-CNN	70-75	7-10	Нет
Mask R-CNN	77	5-8	Да

Вывод

Из проведенного сравнения можно выделить несколько ключевых пунктов, которые подчеркивают основные преимущества и недостатки между рассматриваемыми методами:

1) YOLOv1 базовая модель, демонстрирующая 63% mAP на датасете COCO и высокую скорость обработки 45-50 FPS. Она хорошо подходит для задач реального времени, однако её точность недостаточна для сложных сцен с мелкими объектами или плотными толпами. Модель не поддерживает сегментацию, что ограничивает её применение в задачах, требующих детального выделения объектов.

2) YOLOv5 значительно превосходит YOLOv1 как по точности (mAP 68-73%), так и по скорости (FPS 60-120, в зависимости от варианта модели). Она легко справляется с задачами реального времени и обеспечивает высокую производительность даже на сложных сценах. Однако, как и YOLOv1, она не поддерживает сегментацию, что может быть критично для некоторых задач.

3) SSD достигает mAP 66% и поддерживает скорость 35-45 FPS, что делает её хорошим выбором для задач, требующих баланса между точностью и производительностью. Модель особенно эффективна для мобильных устройств и систем видеонаблюдения. Однако она, как и YOLO, не поддерживает сегментацию, что ограничивает её применение в задачах с высоким уровнем детализации.

4) Fast R-CNN демонстрирует высокую точность (mAP 70-75%) благодаря своей многостадийной архитектуре. Однако низкая скорость

обработки (7-10 FPS) делает её непригодной для задач реального времени. Модель не поддерживает сегментацию, но остаётся полезной для оффлайн-анализа, где приоритетом является точность.

5) Mask R-CNN обеспечивает самую высокую точность среди всех моделей (mAP 77%) и поддерживает сегментацию на уровне пикселей. Это делает её идеальной для сложных задач, таких как медицинская диагностика или анализ изображений с высокой детализацией. Однако её скорость (5-8 FPS) значительно уступает другим моделям, что ограничивает её использование в реальном времени.

Таким образом, выбор модели зависит от требований задачи. Для реального времени предпочтительны YOLOv5 и SSD, для задач сегментации и высокой детализации — Mask R-CNN, а для оффлайн-анализа с упором на точность — Fast R-CNN.

ЗАКЛЮЧЕНИЕ

Выбор модели зависит от требований задачи. Для реального времени предпочтительны YOLOv5 и SSD, для задач сегментации и высокой детализации — Mask R-CNN, а для оффлайн-анализа с упором на точность — Fast R-CNN.

Цель работы, заключавшаяся в классификации методов подсчета количества человека на видео, была достигнута.

Были решены следующие задачи:

1) проведен анализ предметной области методов подсчета количества человека на видео;

2) проведен обзор существующих методов подсчета количества на видео;

3) выделены критерии сравнения рассматриваемых методов и на этом основе классифицированы эти методы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Terven, Juan, Romero-González.* A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas // Machine Learning and Knowledge Extraction 5.4 (2023). — 2024. — С. 1680.
2. *Zhai Shang.* DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion // IEEE access. — 2020. — Т. 8. — С. 24344—24357.
3. *Rice S.* A Fusion Steganographic Algorithm Based on Faster R-CNN. // Computers, Materials & Continua. — 2018. — Т. 55, № 1.
4. *Bharati, Ankita.* Deep learning techniques—R-CNN to mask R-CNN: a survey // Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019. — 2020. — С. 657—668.
5. *Aboyomi D. D., Daniel C.* A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster R-CNN // ITEJ (Information Technology Engineering Journals). — 2023. — Т. 8, № 2. — С. 96—106.

ПРИЛОЖЕНИЕ А

Презентация к научно-исследовательской работе состоит из 6 слайдов