



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 «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

*К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ*

*НА ТЕМУ:*

*«Выявление транспортной структуры на  
изображениях»*

Студент ИУ7-52Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Нисуев Н. Ф.  
(И. О. Фамилия)

Руководитель НИР

\_\_\_\_\_  
(Подпись, дата)

Тассов К. Л.  
(И. О. Фамилия)

*2024 г.*

## РЕФЕРАТ

Расчетно-пояснительная записка к научно-исследовательской работе содержит 25 страниц, 10 иллюстраций, 2 таблицы, 16 источников, 1 приложение.

Научно-исследовательская работа представляет собой изучение предметной области выявления транспортной структуры на изображениях, описание основных методов, а также преимуществ и недостатков каждого из них. Рассмотрены детерминированный подход, экспертный подход и искусственные нейронные сети. Проведено сравнение различных модификаций нейронных сетей: семейства YOLO, семейства R-CNN, а также Single-Shot Detectors.

Ключевые слова: нейрон, перцептрон, сверточная нейронная сеть, капсульная нейронная сеть, R-CNN, SSD, YOLOv7.

# СОДЕРЖАНИЕ

|  |           |
|--|-----------|
| <b>ВВЕДЕНИЕ</b>                              | <b>5</b>  |
| <b>1 Анализ предметной области</b>           | <b>6</b>  |
| 1.1 Детерминированный подход . . . . .       | 6         |
| 1.2 Экспертный подход . . . . .              | 7         |
| 1.3 Нейронные сети . . . . .                 | 8         |
| 1.3.1 Перцептрон . . . . .                   | 9         |
| 1.3.2 Сверточные нейронные сети . . . . .    | 11        |
| 1.3.3 Капсульные нейронные сети . . . . .    | 12        |
| <b>2 Анализ существующих решений</b>         | <b>14</b> |
| 2.1 Семейство нейронных сетей YOLO . . . . . | 15        |
| 2.2 Single Shot Detection . . . . .          | 16        |
| 2.3 R-CNN . . . . .                          | 17        |
| 2.4 Fast R-CNN . . . . .                     | 18        |
| 2.5 Faster R-CNN . . . . .                   | 19        |
| 2.6 Сравнение решений . . . . .              | 22        |
| <b>ЗАКЛЮЧЕНИЕ</b>                            | <b>23</b> |
| <b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>      | <b>25</b> |
| <b>ПРИЛОЖЕНИЕ А</b>                          | <b>26</b> |

## ВВЕДЕНИЕ

Распознавание объектов на изображениях является одной из наиболее распространенных задач при работе с визуальными данными.

Методы распознавания объектов на изображениях находят применение в самых разных областях. В системах управления транспортом они используются для повышения эффективности и безопасности дорожного движения. Видеоматериалы с камер наблюдения анализируются для определения элементов инфраструктуры, таких как перекрестки, туннели, мосты и дороги. В задачах картографирования, для оптимизации потоков транспорта и планирования маршрутов. В оцифровке рукописных текстов, упрощая обработку информации и ее дальнейшее использование.

**Цель научно-исследовательской работы:** анализ существующих методов решения задачи выявления транспортной структуры на изображениях.

Для достижения поставленной цели требуется выполнить следующие задачи:

- рассмотреть предметную область;
- рассмотреть существующие методы решения задачи выявления транспортной структуры на изображениях;
- провести сравнение рассмотренных методов.

# 1 Анализ предметной области

## 1.1 Детерминированный подход

Детерминированный подход основывается на сравнении характеристик распознаваемого объекта с эталонными признаками.

В рамках этого подхода используется множество методов, среди которых наиболее известным является метод контурного анализа [1].

Контурный анализ является совокупностью методов выделения, описания и преобразования контуров изображений и распознавания зрительных образов. Контур целиком определяет форму изображения и содержит всю необходимую информацию для распознавания изображений по их форме. Такой подход позволяет не рассматривать внутренние точки изображения и тем самым значительно сократить объем обрабатываемой информации. Следствием этого является возможность обеспечения работы системы в режиме реального времени.

Основные этапы детерминированного подхода можно разбить на четыре шага:

1. разрывы контура в местах, где яркость меняется не слишком быстро;
2. наличие ложных контуров вследствие шума на изображении;
3. широкие контурные линии из-за размытости или шума.

Преимуществом данного подхода является то, что для поиска и классификации объектов необходимы только эталонные изображения.

В качестве основного недостатка детерминированного подхода выделяют неустойчивость к шумам на используемых изображениях.

## 1.2 Экспертный подход

Экспертный подход основан на разделении множества входных объектов по определенным критериям. Одним из основных методов, использующих данный подход, является дерево решений [2].

Дерево решений представляет собой иерархическую древовидную структуру, в узлах которой содержатся условия, генерируемые в процессе обработки обучающего набора данных. Листья дерева представляют собой классы, по которым производится классификация объектов.

Процесс построения дерева решений включает рекуррентное разбиение обучающего множества на подмножества с применением решающих правил в узлах дерева. Разбиение продолжается до тех пор, пока во всех листьях не окажется конкретный класс, по которому производится классификация. Узел становится листом, когда он содержит объекты одного класса или когда достигается одно из условий остановки, например, максимальная глубина дерева.

Для построения дерева решений обычно применяются жадные алгоритмы, которые выбирают оптимальные решения на каждом шаге алгоритма, но не могут вернуться и изменить их позднее, даже если это было бы оптимальнее для всей системы. На каждом шаге выбирается условие, максимизирующее функцию прироста информации. Для вычисления этой функции используется, например, средняя квадратичная ошибка 1.1.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{\text{true}} - y_{\text{pred}_i})^2, \quad (1.1)$$

где:

- $n$  — число элементов в подмножестве;
- $y_{\text{true}}$  — истинное значение атрибута;
- $y_{\text{pred}_i}$  — предсказанное значение атрибута для  $i$ -го элемента подмножества.

Функция прироста информации может быть выражена зависимостью 1.2.

$$IG = MSE_{\text{root}} - \left( \frac{n_{\text{left}}}{n} MSE_{\text{left}} - \frac{n_{\text{right}}}{n} MSE_{\text{right}} \right), \quad (1.2)$$

где  $MSE_{\text{root}}$  — ошибка в корневом узле,  $MSE_{\text{left}}$  и  $MSE_{\text{right}}$  — ошибки в левом и правом подмножествах соответственно.

Таким образом, на каждом шаге алгоритм выбирает условие, для которого функция прироста информации будет максимальной. Процесс построения дерева останавливается, когда значение функции ошибки равно нулю или когда в узле остается только один элемент, который считается листом.

## 1.3 Нейронные сети

Нейронная сеть - компьютерная программа, которая работает по принципу естественной нейронной сети в мозгу. Задача таких искусственных нейронных сетей - выполнять такие когнитивные функции, как решение проблем и машинное обучение. Теоретические основы нейронных сетей были разработаны в 1943 году нейрофизиологом Уорреном Маккалоком из Университета Иллинойса и математиком Уолтером Питтсом из Чикагского университета. В 1954 году Белмонту Фарли и Уэсли Кларку из Массачусетского технологического института удалось запустить первую простую нейронную сеть. Основная привлекательность нейронных сетей - их способность имитировать умение мозга распознавать образы [3].

Нейронные сети, хотя и не являются точной моделью человеческого мозга, имеют два важных сходства с ним:

- они обучаются, анализируя данные из внешней среды;
- их память и обработка информации основаны на связях между искусственными нейронами.

Благодаря этим характеристикам, нейронные сети обладают рядом преимуществ перед традиционными алгоритмическими системами. Они способны обрабатывать данные параллельно, работать без детально прописанных инструкций и выявлять глобальные закономерности в данных.

Однако такие особенности создают и определенные ограничения: для обучения требуется значительный объем данных, точность вывода не всегда высока, а процесс обучения может быть крайне ресурсоемким.

Существует несколько подходов к построению нейронных сетей, среди которых можно выделить:

- перцептрон;
- сверточные нейронные сети;
- капсульные нейронные сети.

### 1.3.1 Перцептрон

Впервые словосочетание «искусственная нейронная сеть» употребили в статье «Логическое исчисление идей, относящихся к нервной активности». Но развил и воплотил эти идеи в жизнь американский нейрофизик Фрэнк Розенблатт. В 1958 году он предложил схему устройства, моделирующего человеческое восприятие, и назвал его перцептрон (от англ. perception-восприятие). Функция перцептрона заключалась в том, что он должен был передавать информацию от «глаз», представляющих собой фотоэлементы, в блоки электро-механических ячеек памяти, которые по принципам коннективизма абсолютно случайно соединялись между собой. И уже в 1960 в Корнелльском университете был создана и представлена миру физическая модель перцептрона-компьютер Марк 1, который мог различать некоторые буквы английского алфавита на карточках, подносимых к камерам. Основная особенность нейросетей это способность к самообучению и улучшению, уточнению своей функции [4].



Схема элементарного перцептрона представлена на рисунке 1.1.

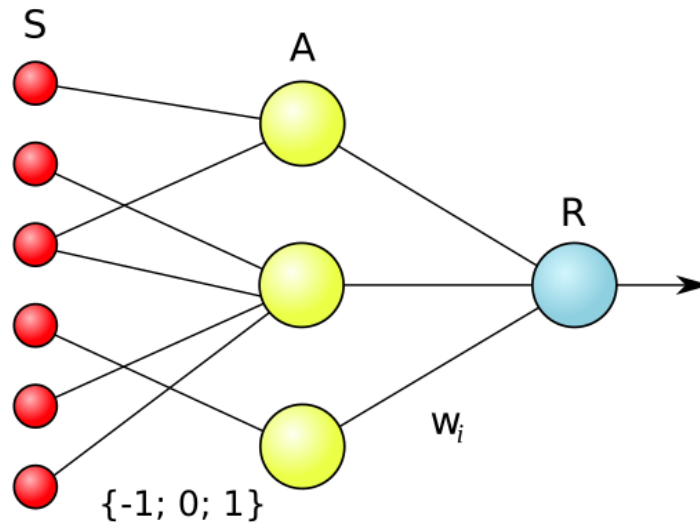


Рисунок 1.1 – Схема элементарного перцептрона

Элементарный перцептрон состоит из элементов трех типов:  $S$ -элементов,  $A$ -элементов и одного  $R$ -элемента.

$S$ -элементы — это слой сенсоров или рецепторов. Каждый рецептор может находиться в одном из двух состояний — покоя или возбуждения, и только в последнем случае он передает единичный сигнал в следующий слой, ассоциативным элементам.

$A$ -элементы называются ассоциативными, потому что каждому такому элементу, как правило, соответствует целый набор  $S$ -элементов.  $A$ -элемент активизируется, как только количество сигналов от  $S$ -элементов на его входе превысило некоторую величину  $\theta$

Сигналы от возбуждившихся  $A$ -элементов, в свою очередь, передаются в сумматор  $R$ , причем сигнал от  $i$ -го ассоциативного элемента передается с коэффициентом  $w_i$ . Этот коэффициент называется весом  $A$ — $R$  связи.

$R$ -элемент подсчитывает сумму значений входных сигналов, помноженных на веса.  $R$ -элемент, а вместе с ним и элементарный перцептрон, выдает 1, если линейная форма превышает порог  $\theta$ , иначе на выходе будет  $-1$ . Математически функцию, реализуемую  $R$ -элементом, можно записать в виде формулы 1.3.

$$f(x) = \text{sign}\left(\sum_{i=1}^n w_i x_i - \theta\right) \quad (1.3)$$

### 1.3.2 Сверточные нейронные сети

Сверточная нейронная сеть (CNN) очень похожа на зрительную кору головного мозга. На зрительной коре имеются небольшие участки клеток нейронов, которые связаны с определенными местами зрительного поля. За это открытие Дэвид Хьюбел и Торстен Визель удостоились Нобелевской премии по медицине 1981 года. Хьюбел и Визель в 1962 году провели эксперимент, в котором показали, что отдельные клетки нейронов откликались исключительно при наблюдении границ конкретной ориентации [5].

Если рассматривать CNN более подробно, то она состоит из серии слоев. Берется изображение, пропускается через чередование сверточных, нелинейных слоев, и с помощью полносвязного слоя порождается вывод. В качестве вывода может выступать класс или вероятность класса, которое лучше всего описывает изображение.

Сверточный слой – это слой, на котором при помощи операции математической свертки заданное количество ядер или окон фильтров в виде матриц также заданного размера проходят по входящей матрице, формируя новую матрицу. Ядро фильтра – матрица весов, которая путем свертки извлекает признаки из пришедших данных. Весовые коэффициенты ядра свертки устанавливаются в процессе обучения и изначально не определены. Фильтр – это объединение нескольких ядер.

Количество фильтров на сверточном слое задается на основе исходной задачи и требований к ее решению: чем фильтров больше, тем выше качество распознавание и ниже быстродействие [6].

Схема сверточной нейронной сети представлена на рисунке 1.2.

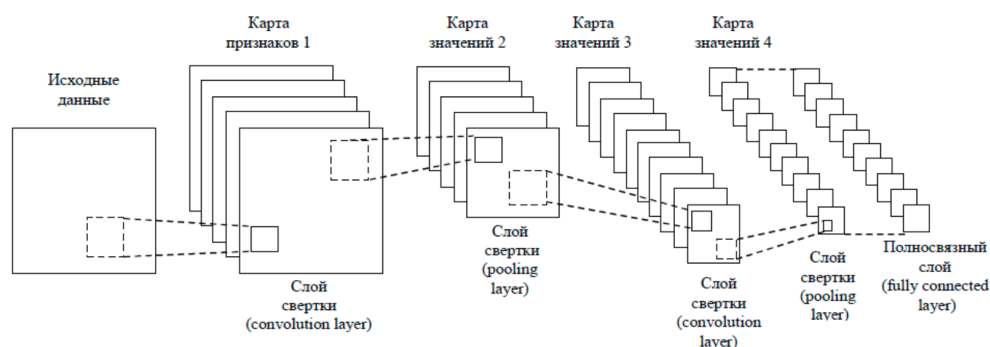


Рисунок 1.2 – Схема сверточной нейронной сети

### 1.3.3 Капсульные нейронные сети

Капсульные нейронные сети содержат так называемый капсульный слой. Капсула — это модифицированный искусственный нейрон, расширенный от скалярной к векторной форме, что позволяет ей хранить больше информации об объекте. На выходе капсулы формируется вектор, способный описывать состояние объекта, например, его позу. Длина вектора определяет вероятность обнаружения объекта, а его направление указывает на состояние объекта [7].

Функция активации в капсулах задается следующим образом:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|},$$

где  $v_j$  — выходной вектор капсулы  $j$ ,  $s_j$  — входной вектор. Правая часть этого выражения делает вектор  $s_j$  единичным, а левая выполняет масштабирование так, что длина  $v_j$  приближается к 1 для больших значений  $s_j$  и к 0 для малых значений.

Для всех капсул, кроме капсул первого слоя, вход  $s_j$  является взвешенной суммой по всем векторам предсказаний  $u_{ji}$  от капсул предыдущего слоя, которые получаются умножением выхода  $u_i$  капсулы нижележащего слоя на весовую матрицу  $W_{ij}$ . Таким образом, вход в капсулу  $j$  определяется выражением:

$$s_j = \sum_i c_{ij} u_{ji},$$

где  $c_{ij}$  — коэффициент связи между капсулами  $i$  и  $j$ , который рассчитывается по формуле:

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k b_{ik}},$$

где  $b_{ij}$  — вероятность того, что капсула  $i$  связана с капсулой  $j$ . Эти вероятности обновляются итеративно, измеряя степень соответствия между текущим выходом  $\mathbf{v}_j$  капсулы  $j$  и предсказанием  $\mathbf{u}_{ji}$  капсулы  $i$ . Соответствие рассчитывается как скалярное произведение  $\mathbf{v}_j$  и  $\mathbf{u}_{ji}$ .

Схема капсульной нейронной сети представлена на рисунке 1.3.

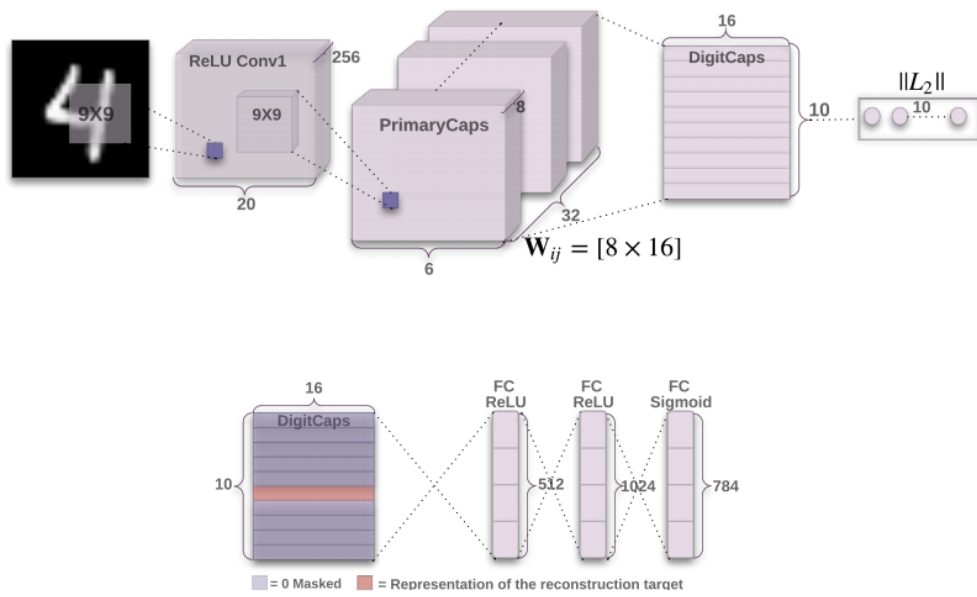


Рисунок 1.3 – Схема капсульной нейронной сети

## 2 Анализ существующих решений

Для решения задачи классификации объектов на изображениях наиболее рационально использовать нейронные сети. Детерминированные методы здесь не подходят, поскольку они неустойчивы к шуму в данных, а метод дерева решений оказывается неприменим из-за отсутствия четких критериев для классификации на этапе построения дерева. Следующим этапом работы станет выбор оптимальной архитектуры нейронной сети.

Полносвязные сети, такие как перцептрон, имеют ограниченное применение в задачах с большим числом входных данных, поскольку сложность их вычислений растет экспоненциально по мере увеличения размеров входных векторов.

Капсульные нейронные сети, хотя и позволяют учитывать пространственное положение объектов, не являются подходящим выбором для данной задачи. Их основное преимущество — выделение иерархических структур в данных — требует значительных вычислительных ресурсов, что делает их менее эффективными для анализа больших изображений. Кроме того, капсульные сети уступают сверточным по устойчивости к шуму, что может отрицательно сказаться на точности классификации [8].

Для поиска и классификации объектов на изображениях существует множество модификаций сверточных нейронных сетей [9], среди которых наиболее популярными являются:

- семейство сетей YOLO;
- Single Shot Detection;
- R-CNN и его модификации.

## 2.1 Семейство нейронных сетей YOLO

YOLO (You Only Look Once) — это алгоритм, который выполняет обнаружение объектов и определение их местоположения на изображении за один проход [10]. YOLO применяет нейронные сети для идентификации объектов в режиме реального времени, и его развитие прошло через несколько версий: от первой YOLOv1 до последующих версий YOLOv2, YOLOv3 и так далее.

Архитектура YOLO построена на основе сверточной нейронной сети (CNN) для выполнения задачи обнаружения объектов. YOLO анализирует изображение целиком, предсказывая ограничивающие рамки и вероятности классов для обнаруженных объектов. Это единая сверточная нейронная сеть, которая одновременно выполняет обнаружение нескольких объектов, определяет их границы и оценивает вероятность принадлежности каждого к определенному классу.

YOLO работает в несколько этапов.

1. Масштабирование изображения до фиксированного размера, например, 416x416, для единообразной обработки.
2. Разделение изображения на сетку  $N \times N$  (например, 13x13) ячеек для локализации объектов.
3. Прогон изображения через сверточную нейронную сеть (CNN), которая извлекает важные признаки для последующей обработки.
4. Прогнозирование нескольких ограничивающих рамок для каждой ячейки сетки, включая координаты, размеры и вероятности классов.
5. Фильтрация рамок по вероятности — рамки с низкой уверенностью отбрасываются.
6. Применение подавления неперекрывающихся рамок (NMS) для устранения наложений и выбора лучших рамок.
7. Возврат окончательных рамок и меток классов для обнаруженных объектов.

Схема YOLO представлена на рисунке 2.1.

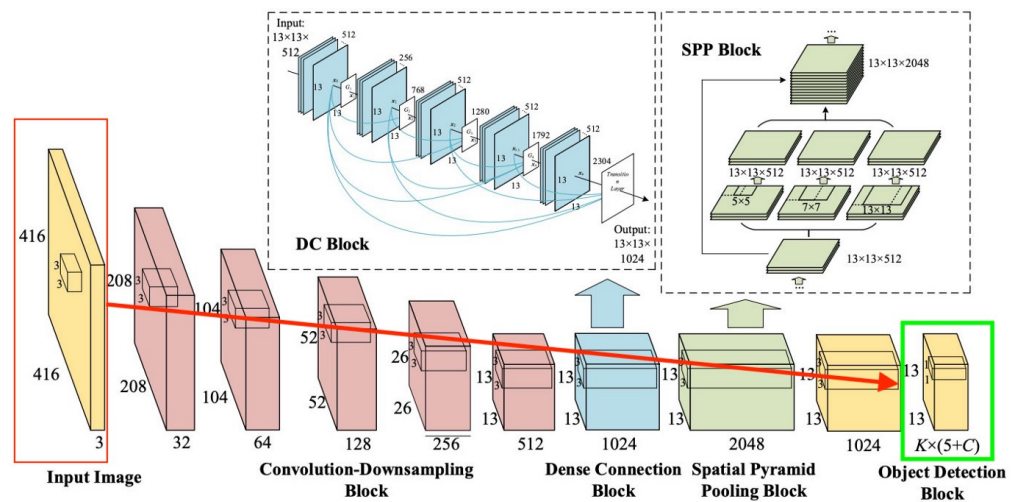


Рисунок 2.1 – Схема YOLO

## 2.2 Single Shot Detection

Single Shot Detection (SSD) использует в своей основе предварительно обученные модели, такие как VGG или ResNet, которые были разработаны для задач классификации на больших наборах данных. На базовую сеть накладываются дополнительные слои, включающие сверточные и пулинговые операции, что позволяет системе выявлять объекты различных размеров [11].

Главное преимущество SSD заключается в её высокой скорости обработки: благодаря тому, что вся работа выполняется одной сетью, она подходит для задач, требующих анализа изображений в реальном времени, таких как системы видеонаблюдения или автопилоты.

Метод SSD эффективно применяет предварительно обученную базовую сеть, что даёт возможность использовать крупные размеченные датасеты, изначально созданные для задач классификации. Однако этот подход имеет свои недостатки. Одной из ключевых проблем является чувствительность к размерам объектов: если их масштаб существенно отличается от объектов, представленных в обучающем наборе, точность обнаружения может снижаться. Это связано с тем, что дополнительные слои сети оптимизированы для обработки объектов лишь в определённых диапазонах размеров.

Схема SSD представлена на рисунке 2.2.

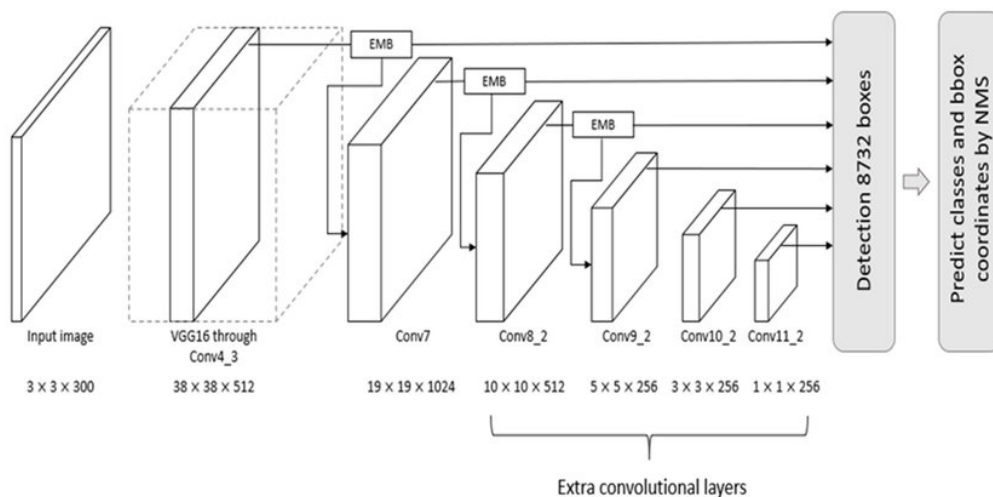


Рисунок 2.2 – Схема SSD

## 2.3 R-CNN

R-CNN (Regions with Convolutional Neural Networks) — это метод обнаружения объектов, разработанный командой из UC Berkley, который работает, выделяя потенциальные области, где могут находиться объекты, и применяя сверточные нейронные сети к этим областям для классификации и локализации объектов [12]. Процесс можно разделить на несколько основных шагов.

1. R-CNN сначала применяет алгоритм, такой как Selective Search, чтобы сгенерировать около 2000 регионов-кандидатов.
2. Каждая из этих областей изменяется под фиксированный размер, чтобы затем передать в нейронную сеть для обработки.
3. Каждый выделенный регион проходит через сверточную нейронную сеть. На этом этапе R-CNN определяет, какой объект находится в регионе, и создает ограничивающую рамку для точного выделения его позиции.
4. Для повышения точности рамки дополнительно корректируются с использованием регрессионной модели, чтобы максимально соответствовать границам объекта.
5. На основе предсказанных классов и рамок, R-CNN отфильтровывает регионы, чтобы получить окончательные результаты.



Схема R-CNN представлена на рисунке 2.3.

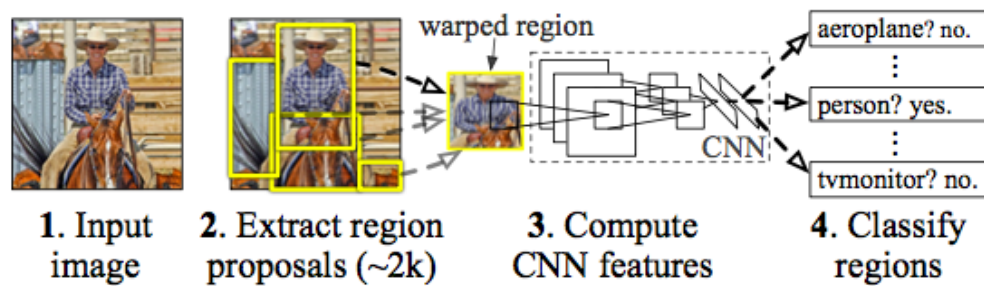


Рисунок 2.3 – Схема R-CNN

Метод R-CNN обеспечивает хорошее качество обнаружения, но имеет ряд недостатков:

- Поскольку R-CNN использует разные модели для каждого этапа, каждая из них обучается отдельно, что может приводить к несовместимым результатам;
- R-CNN обрабатывает каждый регион-кандидат отдельно, что требует значительных вычислительных ресурсов и времени;
- R-CNN требует нескольких этапов обучения: сначала обучается модель для выделения регионов, затем модель для классификации, и дополнительно — для регрессии ограничивающих рамок.

## 2.4 Fast R-CNN

Fast R-CNN была разработана для преодоления недостатков медлительности и сложности обучения, характерных для R-CNN [13]. В отличие от R-CNN, где для каждого региона-кандидата выполнялось отдельное прохождение сети, в Fast R-CNN все изображение пропускается через сверточные слои один раз. Это позволяет эффективно извлекать признаки для всего изображения сразу. После извлечения признаков для всего изображения, Fast R-CNN использует RoI-пулинг, чтобы выделить соответствующие регионы-кандидаты (регион интереса, RoI). RoI-пулинг преобразует размер этих регионов в фиксированный, что позволяет обрабатывать их в одном этапе. Каждый

регион-кандидат классифицируется в один из классов объектов, а также предсказывается точная позиция ограничивающей рамки для каждого объекта. Эти два этапа выполняются параллельно, что ускоряет обработку.

Схема Fast R-CNN представлена на рисунке 2.4.

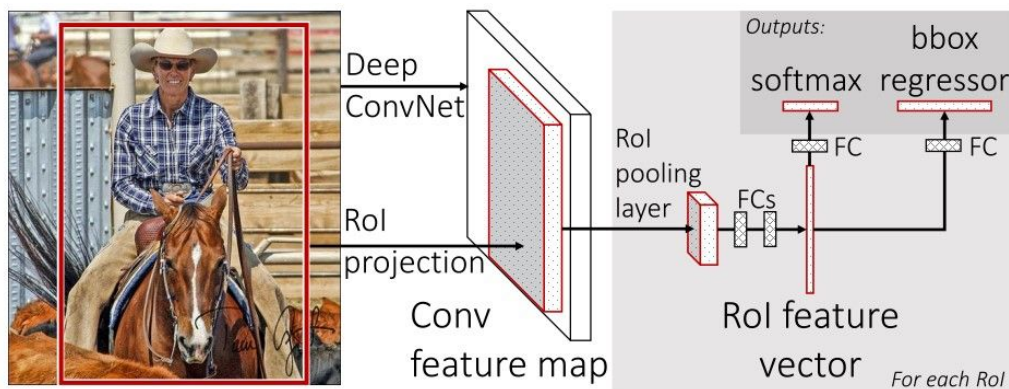


Рисунок 2.4 – Схема Fast R-CNN

Основная причина, по которой Fast R-CNN работает быстрее, чем R-CNN, заключается в том, что сверточная операция выполняется только один раз для каждого изображения, а не для 2000 регионов-кандидатов. Это позволяет сразу получить карту признаков, из которой выделяются необходимые регионы.

Fast R-CNN значительно ускоряет процесс обучения и тестирования по сравнению с R-CNN. Тем не менее, генерация регионов-кандидатов продолжает быть основным ограничивающим фактором, который сдерживает производительность алгоритма.

## 2.5 Faster R-CNN

Оба упомянутых алгоритма, R-CNN и Fast R-CNN, используют метод выборочного поиска для формирования регионов-кандидатов, что является медленным и требовательным к ресурсам процессом, ограничивающим общую производительность сети. Чтобы решить эту проблему, Шаочин Рен и коллеги предложили алгоритм обнаружения объектов, который исключает выборочный поиск, позволяя сети самостоятельно генерировать региональные предложения [14].

В алгоритме Faster R-CNN, как и в Fast R-CNN, изображение сначала пропускается через сверточную сеть для создания карты признаков. Однако,

вместо применения медленного выборочного поиска для выделения регионов-кандидатов, Faster R-CNN использует дополнительную сеть — Region Proposal Network (RPN), которая эффективно предсказывает области, вероятно содержащие объекты. Далее, с помощью слоя Region of Interest (RoI) Pooling, эти региональные предложения адаптируются для классификации объектов и уточнения координат ограничивающих рамок. Этот подход делает процесс выделения областей намного более быстрым и точным, что позволяет обрабатывать изображения в реальном времени и обеспечивает высокую эффективность модели в задачах обнаружения объектов.

Схема Faster R-CNN представлена на рисунке 2.5.

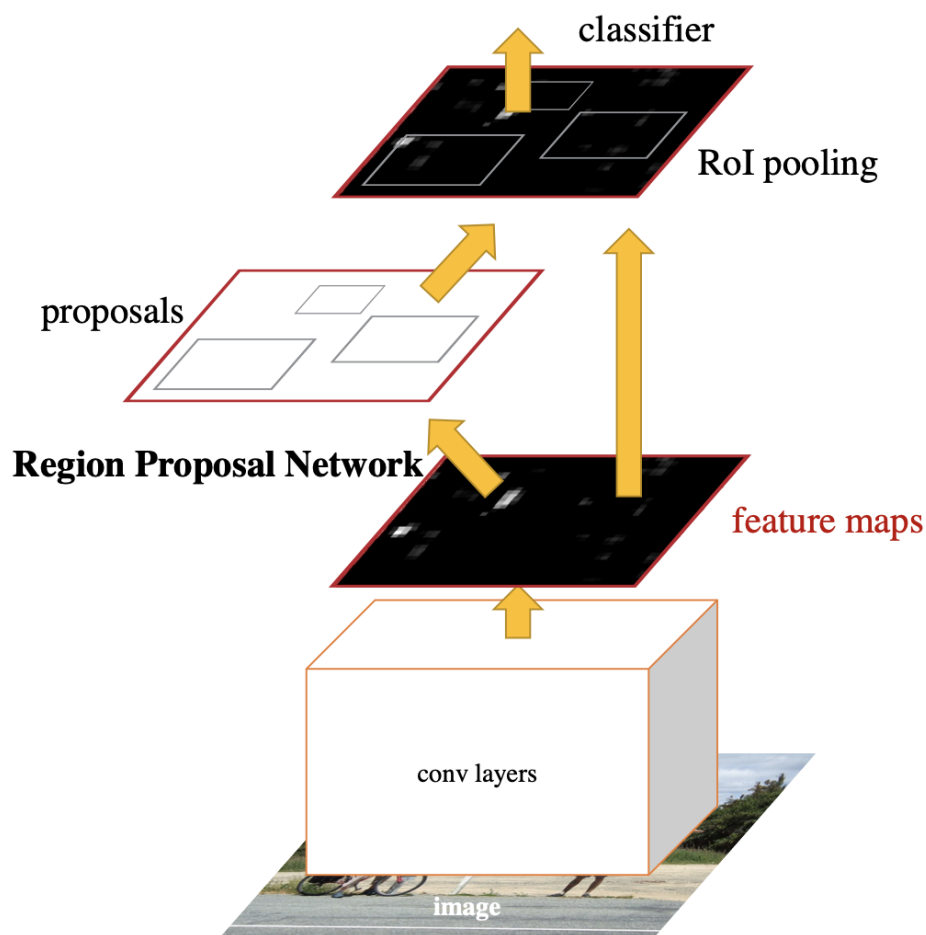


Рисунок 2.5 – Схема Fast R-CNN

Сравнение скорости R-CNN, Fast R-CNN и Faster R-CNN представлено на рисунке 2.6 [15].

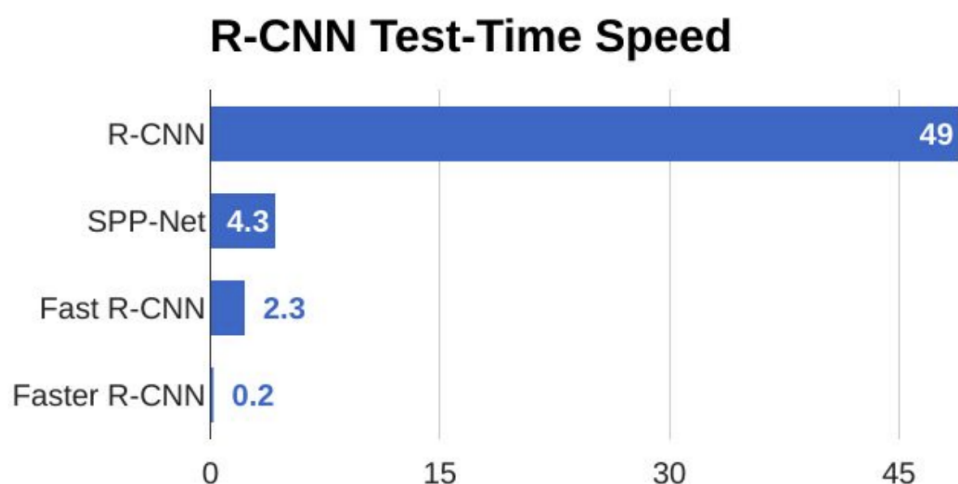


Рисунок 2.6 – Сравнение скорости R-CNN, Fast R-CNN и Faster R-CNN

Сравнение средней точности и скорости обработки изображений на разных наборах данных R-CNN, Fast R-CNN, Faster R-CNN представлено на таблице 2.1 [16].

Таблица 2.1 – Сравнение R-CNN, Fast R-CNN и Faster R-CNN

| Модель       | Набор данных | Средняя точность (%) | FPS   |
|--------------|--------------|----------------------|-------|
| RCNN         | ILSVRC2013   | 31.4                 | 0.077 |
| RCNN         | VOC 2010     | 53.7                 | 0.077 |
| Fast R-CNN   | VOC 2007     | 70                   | 3.33  |
| Fast R-CNN   | VOC 2010     | 68.8                 | 3.33  |
| Fast R-CNN   | VOC 2012     | 68.4                 | 3.33  |
| Faster R-CNN | VOC 2007     | 78.8                 | 5     |
| Faster R-CNN | VOC 2012     | 75.9                 | 5     |
| Faster R-CNN | COCO         | 42.1                 | 5     |

На основе таблицы 2.1 и рисунка 2.6 можно сделать вывод что Faster R-CNN быстрее и точнее своих прдшественников. Исходя из этого дальше будет рассматриваться только Faster R-CNN.

## 2.6 Сравнение решений

Рассмотренные выше решения были протестированы на скорость и точность работы на наборе данных COCO [16].

Результаты сравнения существующих решений представлены в таблице 2.2.

Таблица 2.2 – Сравнение существующих решений

|                          | <b>Faster R-CNN</b> | <b>SSD</b> | <b>YOLOv7</b> |
|--------------------------|---------------------|------------|---------------|
| Скорость (FPS)           | 5                   | 19         | 161           |
| Средняя точность (%)     | 42.1                | 48.5       | 69.7          |
| Сложность обучения       | Средняя             | Средняя    | Низкая        |
| Устойчивость к масштабам | +                   | +          | +             |

Из таблицы 2.2 можно сделать вывод, что YOLOv7 точнее и быстрее чем Faster R-CNN и SSD, а также легче в обучении.

## ЗАКЛЮЧЕНИЕ

Определить универсально лучший метод распознавания объектов на изображениях невозможно, так как выбор подходящего подхода зависит от особенностей конкретной задачи. Современные исследования в этой области преимущественно направлены на адаптацию и усовершенствование существующих технологий под специализированные задачи.

**Цель научно-исследовательской работы достигнута:** проанализированы существующие методы решения задачи выявления транспортной структуры на изображениях.

В ходе выполнения данной научно-исследовательской работы были решены следующие задачи:

- рассмотрены предметную область;
- рассмотрены существующие методы решения задачи выявления транспортной структуры на изображениях;
- проведено сравнение рассмотренных методов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Сакович И. О., Белов Ю. С.* Обзор основных методов контурного анализа для выделения контуров движущихся объектов // Инженерный журнал: наука и инновации. — 2014. — 12 (36). — С. 11.
2. *Мифтахова А. А.* Применение метода дерева решений для решения задач классификации и прогнозирования // Инфокоммуникационные технологии. — 2016. — Т. 14, № 1. — С. 64—70.
3. *Ксенофонтов В. В.* Нейронные сети // Проблемы науки. — 2020. — № 11. — С. 59.
4. *Вовченко Р., Гайворонский Г., Беляев Н.* ПЕРЦЕПТРОН КАК ОСНОВА СОВЕРЕМЕННЫХ ИНС // Новая наука: Теоретический и практический взгляд. — 2016. — № 117—2. — С. 129—131.
5. *Романов А. А.* Сверточные нейронные сети // Научные исследования: ключевые проблемы III тысячелетия. — 2018. — С. 5—9.
6. *Багаев И. И.* Анализ понятий нейронная сеть и сверточная нейронная сеть, обучение сверточной нейросети при помощи модуля TensorFlow // Математическое и программное обеспечение систем в промышленной и социальной сферах. — 2020. — Т. 8, № 1. — С. 15.
7. *Зубричев Н. В.* Новый вид нейронной сети для распознавания изображений-капсульная нейронная сеть // Современные технологии: актуальные вопросы, достижения и инновации. — 2017. — С. 106—108.
8. *Морзавченков А.* КАПСУЛЬНЫЕ НЕЙРОННЫЕ СЕТИ В ЗАДАЧАХ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ // ВСЕРОССИЙСКАЯ СТУДЕНЧЕСКАЯ КОНФЕРЕНЦИЯ «СТУДЕНЧЕСКАЯ НАУЧНАЯ ВЕСНА», ПОСВЯЩЕННАЯ 85-ЛЕТИЮ ЮА ГАГАРИНА. — 2019. — С. 150—151.
9. *Резунков Д. А., Шорникова Н. Ю.* СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФРЕЙМВОРКОВ И НЕЙРОСЕТЕВЫХ АРХИТЕКТУР ДЛЯ КОМПЬЮТЕРНОГО ЗРЕНИЯ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ОПРЕДЕЛЕНИЯ ОБЪЕКТОВ НА АЭРОСЪЕМКАХ // Педагогика, психология и экономика: вызовы современности и тенденции развития. — 2024. — С. 100.

10. *Стаценко С., Афанасьева Ж.* РАСПОЗНАВАНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ С ПОМОЩЬЮ YOLO АРХИТЕКТУРЫ // Электронный научный журнал "Молодая наука Сибири". — 2023. — 2 (20).
11. *Петухов Д. Е., Ткаченко А. В., Белов Ю. С.* ОБНАРУЖЕНИЕ ОБЪЕКТОВ С ПОМОЩЬЮ SSD (SINGLE SHOT ДЕТЕКТОР) // Сборник избранных статей по материалам научных конференций ГНИИ "Нацразвитие". — 2020. — С. 166—170.
12. Oriented R-CNN for object detection / X. Xie [и др.] // Proceedings of the IEEE/CVF international conference on computer vision. — 2021. — С. 3520—3529.
13. *Сирота А. А., Митрофанова Е. Ю., Милованова А. И.* Анализ алгоритмов поиска объектов на изображениях с использованием различных модификаций сверточных нейронных сетей // Вестник ВГУ. Серия: Системный анализ и информационные технологии. — 2019. — № 3. — С. 123—137.
14. *Ren S.* Faster r-cnn: Towards real-time object detection with region proposal networks. — 2015.
15. *Ghandi R.* R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms // Towards Data Science. — 2022.
16. *Rashid T. M. N. B. T., Fadzil L. M.* Comparative Review of Object Detection Algorithms in Small Single-Board Computers // International Journal on Recent and Innovation Trends in Computing and Communication. — 2023.



## **ПРИЛОЖЕНИЕ А**

### **Презентация к научно-исследовательской работе**

Презентация содержит 3 слайда.