РЕФЕРАТ

Расчетно-пояснительная записка к научно-исследовательской работе содержит 32 страницы, 12 иллюстраций, 2 таблицы, 9 источников, 1 приложение.

Научно-исследовательская работа посвящена изучению методов идентификации и отслеживания объектов на видеопотоках. Рассматриваются основные подходы к решению задачи идентификации, использующие искусственные нейронные сети. Приводится сравнение алгоритмов идентификации и отслеживания по различным критериям.

Ключевые слова: видеонаблюдение, идентификация объектов, отслеживание объектов, нейронные сети, YOLO, R-CNN, RetinaNet, Deep-SORT, прогнозирование траекторий.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Системы видеонаблюдения и анализа видеоинформации стали важной частью современной технологии осуществления безопасности. Одной из актуальных задач становится автоматизированная идентификация объектов на видеопотоках и их дальнейшее отслеживание. Решение данной задачи особенно востребовано в системах, связанных с безопасностью, мониторингом общественных мест, дорог. В условиях наблюдения объектов в динамической среде одной лишь идентификации объекта оказывается недостаточно. Возникает необходимость не только идентифицировать объект, но и прогнозировать его возможное местоположение на последующих кадрах или на других видеокамерах. Такой подход позволяет продолжить отслеживание объекта, даже если он временно исчезает из поля зрения одной камеры, но может вскоре появиться в зоне видимости другой. Следовательно, эффективные методы идентификации и прогнозирования траекторий движения объектов оказываются крайне значимыми для повышения надежности и точности систем наблюдения.

Целью данной работы является анализ существующих методов идентификации объектов на видео, а так же методов их отслеживания. Для достижения поставленной цели необходимо выполнить следующие задачи:

1) Изучить современные методы идентификации объектов на изображениях и видео;
2) Сравнить методы идентификации объетов по выбранным критериям;
3) Изучить методы отслеживания объектов на кадрах видеопотоков;
4) Сравнить методы отслеживания объектов на кадрах видеопотоков.

1 Обзор предметной области

Задачей распознавания объекта называется построение алгоритма, который вычисляет некоторые характеристики это объекта по его наблюдаемым свойствам [4]. Работать этот алгоритм должен не только для объектов, предьявленных заранее, но и для объектов, которые заранее представлены не были (рисунок 1.1). Задачей обучения является построение таких алгоритмов по имеющемуся набору объектов.

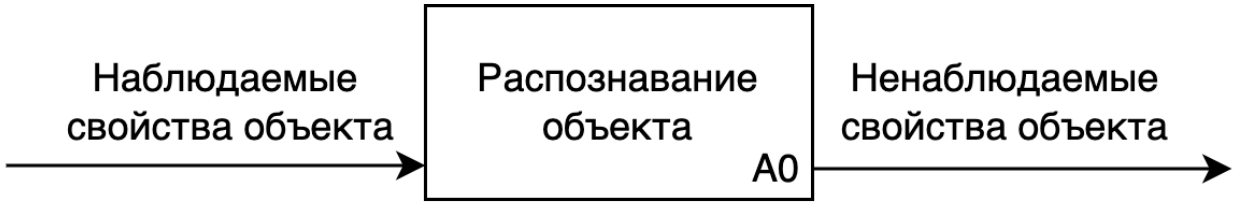

Рисунок 1.1 – Схема алгоритма, решающего задачу распознавания объектов в формате `idef0`

Под наблюдаемыми свойствами объекта принимаются значения векторов свойств, образующих некоторое пространство свойств X. Аналогично, результаты распознавания являются результаты векторов в пространстве ответов Y. Исходя из этого, алгоритм решающий задачу распознавания объекта осуществляет некоторое отображения $X \to Y$.

Можно привести такой пример:

Пусть имеется некоторый обучающий набор из n объектов с известными наблюдаемыми признаками из X и известными ненаблюдаемыми признаками из Y (формула 1.1).

$$M = ((x_1, y_1), ..., (x_n, y_n) : x_i \in X, y_i \in Y) \tag{1.1}$$

В качестве результата алгоритм распознавания для некоторого объекта O с наблюдаемыми свойствами x будет выдавать результат y_i такой, что наиболее близким значением наблюдаемых свойств к x (согласно какой-то метрике) будет значение наблюдаемых свойств x_i.

Так же необходимо формализовать постановку задачи обучения. Пусть имеется пространство наблюдаемых свойств X, пространство ненаблюдаемых

свойств Y, пространство алгоритмов распознавания $A : X \to Y$, пространство вероятностных мер P на $X \times Y$, функция штрафа L (формула 1.2), обучающий набор M.

$$L(a(x), y, x) = \begin{cases} 0, & \text{если } a(x) = y, \\ 1, & \text{если } a(x) \neq y. \end{cases} \tag{1.2}$$

Требуется по этим данным, построить алгоритм $a \in A$, при котором математическое ожидание штрафа минимально по некоторому распределению $\pi \in P$ (формула 1.3).

$$E_p(f) = \int_{x,y} E(a(x), y, x)\, d\pi(x, y) \to \min_a \tag{1.3}$$

По методу Монте-Карло, можно приблизить математическое ожидание штрафа (формула 1.4).

$$E(a, M) = \frac{1}{N} \sum_{i=1}^{N} E(f(x_i, y_i, x_i) \to \min_a . \tag{1.4}$$

Приведенный способ называется способом минимизации ошибки обучения. В нем есть такой недостаток: в результате обучения был составлен некоторый распознающий алгоритм a такой, что $a(x_i) = y_i$, и он имеет малую ошибку на обучающем наборе и большую ошибку на случайных значениях. Такая ситуация называется переобучением.

В качестве примера алгоритма распознавания можно привести распознающие деревья. Для распознаваемого объекта проводится некоторая конечная цепочка сравнений значений его наблюдаемых свойств с некоторыми значениями. Обучение дерева заключается в составлении его структуры, значений и операций для сравнения и ответов в каждом листе. Пусть имеем дерево с одним листом ($f(x) = r$). При квадратичной ошибке минимум по r достигается в среднем арифметическом ответе из всего обучающего набора (формула 1.5).

$$r = \frac{1}{N} \sum_{i=1} N y_i \tag{1.5}$$

Во многих реальных задачах размерность пространства наблюдаемых свойств не совсем естественна. Чаще всего такие объекты представляются ввиде какого-то структурированного набора из более элементарных объектов.

Эти элементарные объекты уже можно закодировать вектором свойств. Изображение представленно в виде набора пикселей, которые в свою очередь могут кодироваться некоторым вектором наблюдаемых свойств.

Вероятностной моделью последовательностей в пространстве X называется последовательность совместных распределений вероятности $p^k(x_1, ..., x_k)$ (формула 1.6).

$$p^k(x_1, ..., x_k) = p_{k|k-1}(x_k|x_1, ..., x_{k-1}) \cdot p^{k-1}(x_1, ..., x_{k-1}) = \\ = ... = p_{k|k-1}(x_k|x_1, ..., x_{k-1})...p_{2|1}(x_2|x_1) \cdot p^1(x_1) \quad (1.6)$$

В модели Маркова условные вероятности $p_{k|k-1}$ зависят от фиксированного числа величин, имеющими больший коэффициент (наиболее близкие к k). То есть вероятность следующей величины зависит от вероятностей n предыдущих величин. На рисунках 1.2, 1.3 приведены схемы зависимости вероятностей случайных величин (в данном случае свойств), для моделей Маркова разной степени. В круге представлены вероятности, стрелки определяют направление зависимости вероятности свойства (к зависимому).

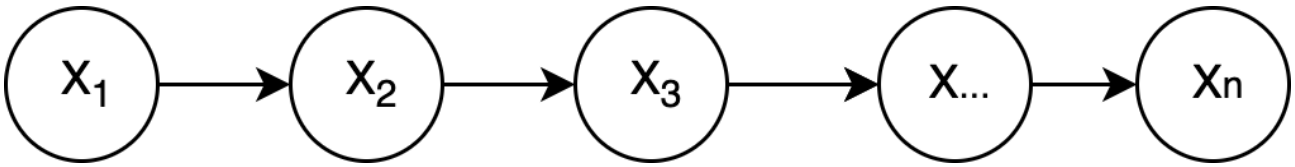

Рисунок 1.2 – Зависимости случайных свойств в модели Маркова, при n = 1

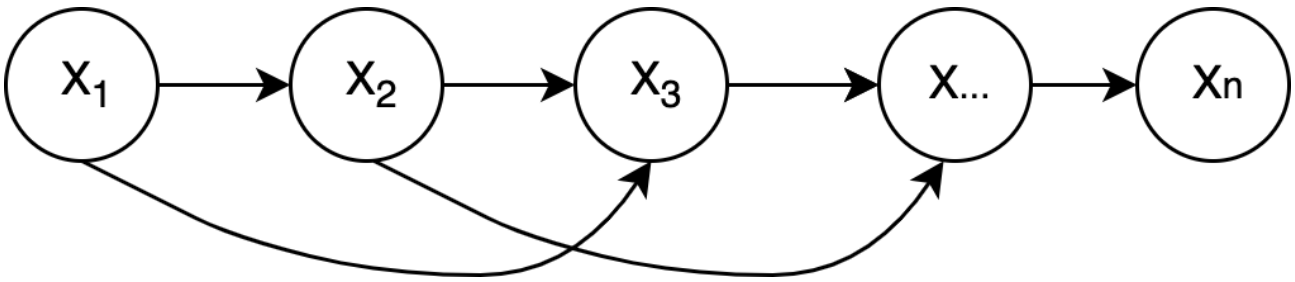

Рисунок 1.3 – Зависимости случайных свойств в модели Маркова, при n = 2

На рисунке 1.4 представлена схема зависимости вероятности величин (в данном случае и наблюдаемых, и ненаблюдаемых свойств) в скрытой

модели Маркова (HMM). В ней текущее состояние системы (ненаблюдаемые свойства) неизвестно напрямую, а вместо него видно лишь определенные состояния (наблюдаемые свойства), которые зависят от скрытых состояний. С каждым скрытым состоянием связана вероятность наблюдения определенного открытого состояния. При работе со скрытой моделью Маркова приходится решать задачу нахождения скрытых состояний на основе видимых состояний. Эту задачу как раз решает алгоритм распознавания $A : X \to Y$.

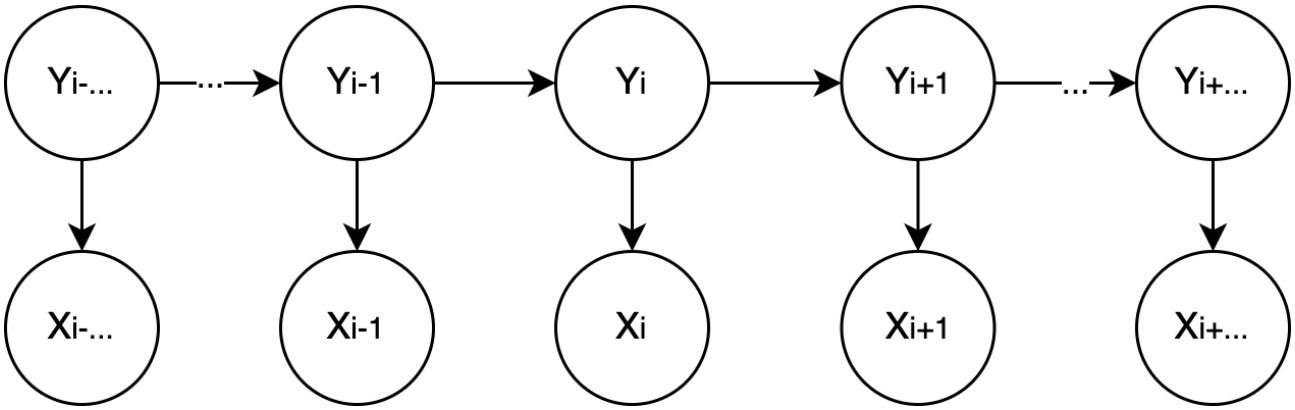

Рисунок 1.4 – Зависимости случайных свойств в модели Маркова, при n = 2

Таким образом, обычная марковская модель применима в задачах отслеживания некоторой последовательности действий. Это как раз подходит для отслеживания объекта и его действий после идентификации. Скрытая марковская модель может быть полезна, если приходится решать задачу распознавания. В задаче распознавания объекта на видеопотоках, скрытые состояния могут представлять собой сам класс объекта или что-то, что его идентифицирует, а так же его позицию (относительно того, на какой камере его можно увидеть). Открытыми состояниями могут быть наблюдаемые свойства объекта, такие как форма, цвет, движения. Эти наблюдения зависят от скрытых состояний и могут быть неполными, но модель может интерпретировать их в зависимости от переходных вероятностей. Модель также может учитывать, что объект на основе текущего состояния (например скорости) будет двигаться плавно (или наоборот) и появится на определенной камере. Это может быть особенно полезно в ситуациях, когда видеокамеры не перекрывают области видимости друг друга и объект в данный момент находится вне поле зрения.

2 Сверточные нейронные сети

Сверточная нейронная сеть состоит из следующих слоев: сверточные слои, субдискретизирующие слои, слои перцептрона. Задача сверточных и субдискретизирующих слоев состоит в том, чтобы формировать входной вектор наблюдаемых свойств, который будет передан перцептрону. В контексте обработки изображений, свертка — это процесс, при котором фильтр (или ядро) применяется к изображению, чтобы извлечь из него наблюдаемые признаки. Это делается путем перемещения фильтра по изображению и вычисления взвешенной суммы пикселей, которые фильтр охватывает. Для положения фильтра вычисляется сумма 2.1, где N, M - размеры фильтра, p - пиксель изображения, w - вес. Эта сумма образует новый пиксель, который затем становится частью выходного изображения.

$$S = \sum_{i=0,j=0}^{i=N,j=M} p_{ij} \cdot w_{ij} \tag{2.1}$$

Свертка позволяет извлекать такие наблюдаемые признаки с изображения как края и текстуры. Каждый фильтр может извлекать какой-то один наблюдаемый признак, поэтому для формирования вектора наблюдаемых признаков необходимо иметь несколько фильтров, каждый из которых будет извлекать конкретный признак. На рисунке 2.1 представлена схема прохода фильтра размером 3x3 по изображению 5x5.

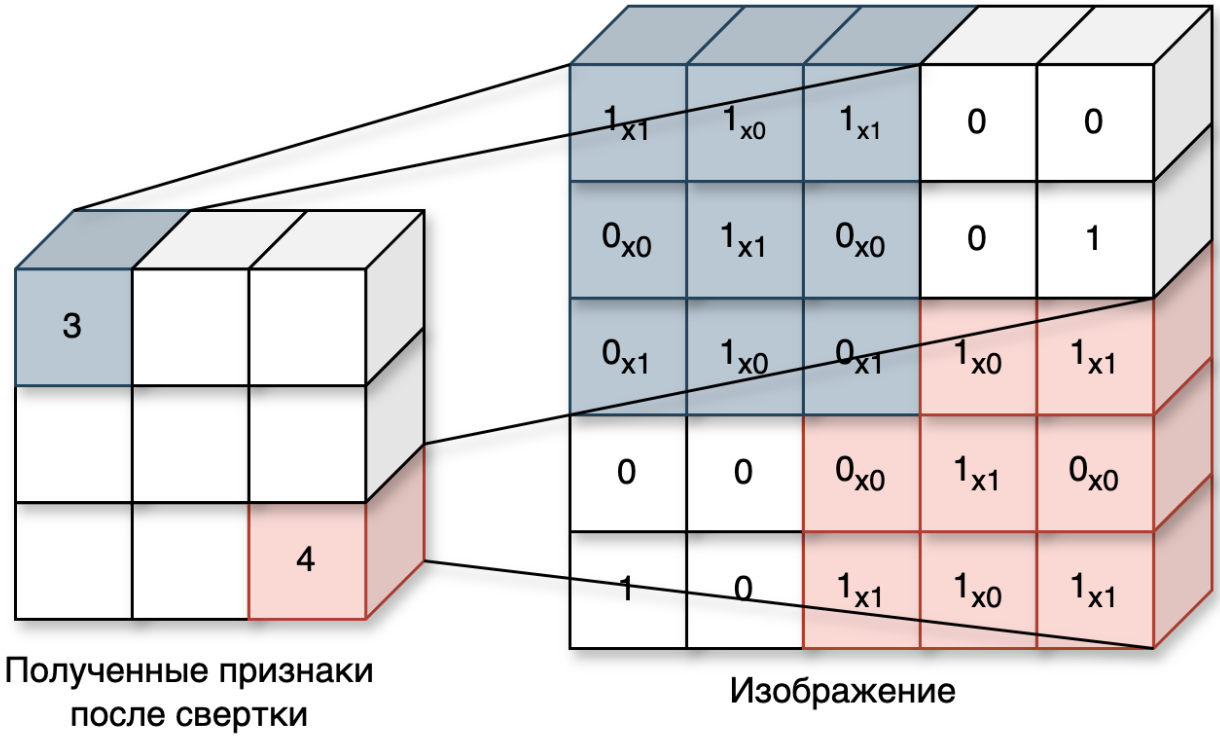

Рисунок 2.1 – Схема прохода фильтра размеров 3x3 по изображению 5x5

В представленной схеме есть недостаток. Крайние пиксели изображения никогда не оказываются в центре ядра, так как тогда ядру будет неоткуда брать информацию из пикселей рядом с крайним вне изображения. Эту проблему решает технология padding. Ее суть заключается в том, чтобы прибавить к изображению ложные пиксели нулевого значения. На рисунке 2.2 представлена схема прохода фильтра по изображению с ложными пикселями нулевого значения.

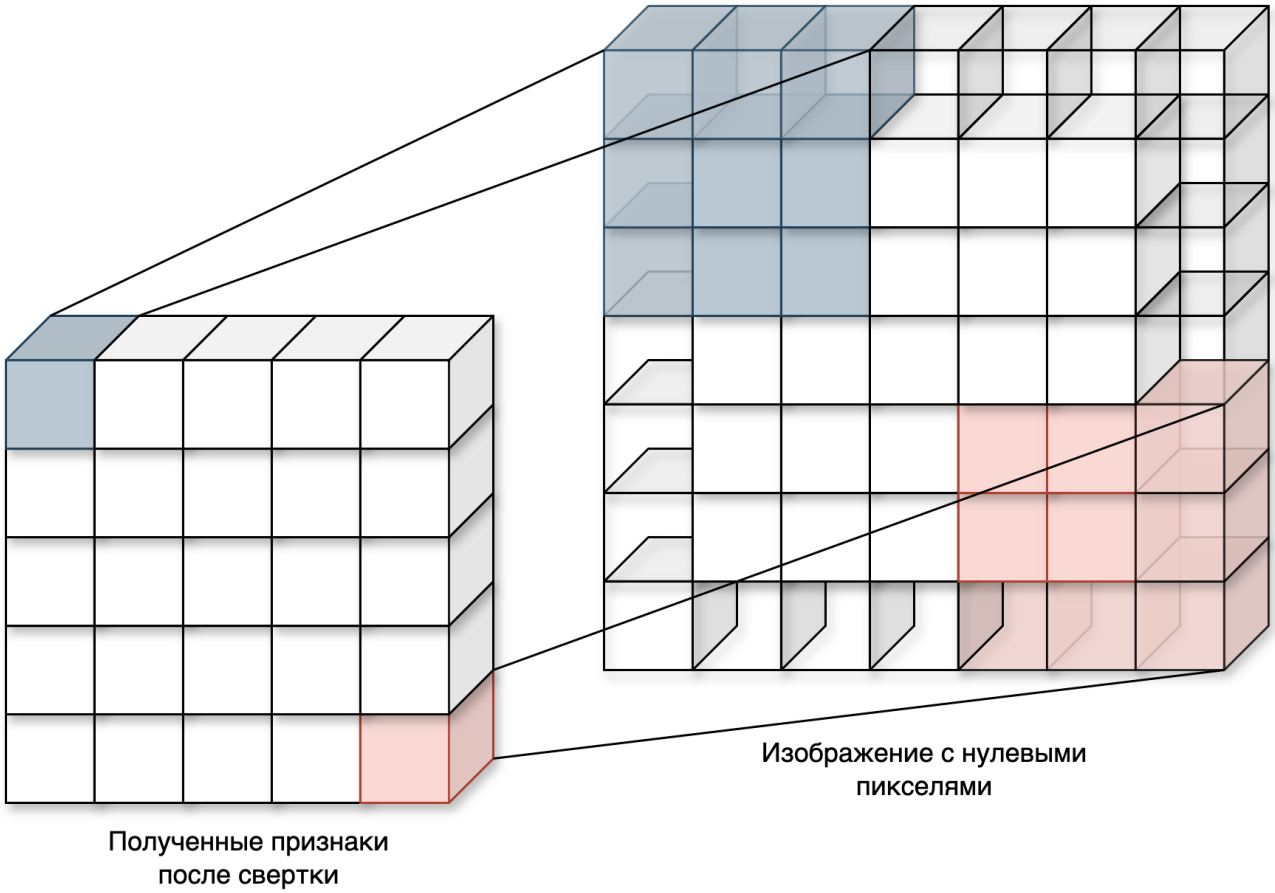

Рисунок 2.2 – Схема прохода фильтра по изображению с ложными пикселями

Субдискретизирующие слои уменьшает размерность матриц, полученных на этапе свертки. На этом этапе фильтр ”скользит”вдоль матрицы, полученной на этапе свертки и выполняет либо усреднение (average pooling) или выбор максимального (max pooling) из сканируемой области.

Таким образом, сверточные слои можно располагать друг за другом, формируя иерархию признаков - от низкоуровневых (края и текстуры), которые выделяются в начальных слоях, до высокоуровневых (формы), которые выделяются в более глубоких слоях. Подвыборочные (субдискретизирующие) слои обычно следуют за одним или несколькими сверточными слоями и являются промежуточными шагами для уменьшения размерности, подготавливая данные к более глубоким сверточным слоям или к полносвязанным слоям.

Полносвязный слой (или перцептрон) занимается классификацией признаков, полученных от подвыборочных слоев. Каждый подвыборочный слой связан с одним нейроном полносвязного слоя. Значение нейрона

вычисляется по формуле 2.2, где X это вектор подготовленных наблюдаемых свойств, переданных нейрону, размерностью X, w - веса, b - коэффициент сдвига слоя, f - функция активации.

$$Y = f(\sum_{i}^{N} X_i \cdot w_i + b) \tag{2.2}$$

3 Рекуррентные нейронные сети

Рекуррентные нейронные сети предназначены для обработки последовательности данных, серий событий, например распознавание речи, текста. Ключевая особенность рекуррентных нейронных сетей состоит в наличии обратных связей, которые позволяют учитывать предыдущие состояния сети при обработке текущего набора признаков. Таким образом, говорят, что рекуррентные нейронные сети обладают памятью или запоминанием контекста. В формуле 3.1 представлено условное выражение для определения текущего скрытого признака y_i в момент времени i, учитывая результат определения скрытого признака y_{i-1} в момент времени $i-1$.

$$y_i = f(W_y \cdot y_{i-1} + W_x \cdot x_i + b) \tag{3.1}$$

На рисунке 3.1 представлена общая схема работы рекуррентной нейронной сети и ее развертка, где X - вектор временных признаков, Y результат, y_i - результат на каждом i-том шаге.

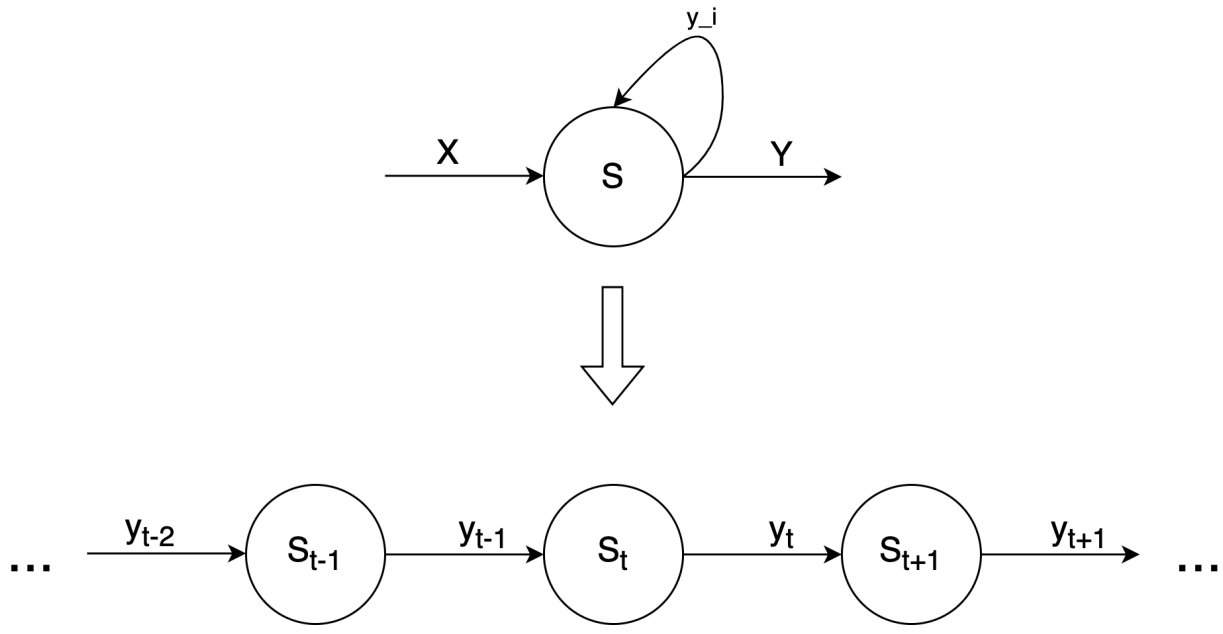

Рисунок 3.1 – Схема работы РНС

4 Распознавание и детектирование объектов

В предыдущей главе были представлены сверточные нейронные сети, с помощью которых можно решать задачу распознавания изображения. Для этого необходимо иметь ядра (фильтры) для выявления наблюдаемых признаков из изображения. Получить их можно в результате обучения. В начале обучения значения весов ядра случайны, в процессе обучения они корректируются, чтобы уменьшить ошибку предсказания. Изображение проходит через сеть, ядра свёртки создают карты признаков путём свертки, в конце прохода сеть вычисляет ошибку предсказания. Модель использует алгоритм обратного распространения для корректировки весов в ядрах на основе ошибки. Вес фильтров, которые лучше помогают отличать признаки объектов, настраиваются так, чтобы усилить эти признаки, а менее значимые фильтры корректируются или игнорируются. На начальных слоях СНС учится выделять простые признаки, такие как линии и углы, что помогает определить границы объектов. На более глубоких уровнях сети активируются более сложные признаки, которые представляют собой комбинации линий и текстур, характерных для форм объектов.

Для решения задачи распознавания объектов необходимо составить архитектуру нейронной сети.

4.1 LeNet-5

Архитектура LeNet-5 была создана для распознавания рукописных цифр, однако она может быть адаптирована для распознавания объектов. Ее структура состоит в следующем:

1) Входное изображение.
2) Сверточный слой с 6 фильтрами 5x5.
3) Подвыборочный слой Max Pooling 2x2.
4) Сверточный слой с 16 фильтрами 5x5.
5) Подвыборочный слой Max Pooling 2x2.

6) Полносвязный слой с 120 нейронами.

7) Выходной слой Softmax для классификации на основе признаков лица.

На рисунке 4.1 представлена визуализация архитектуры LeNet–5.

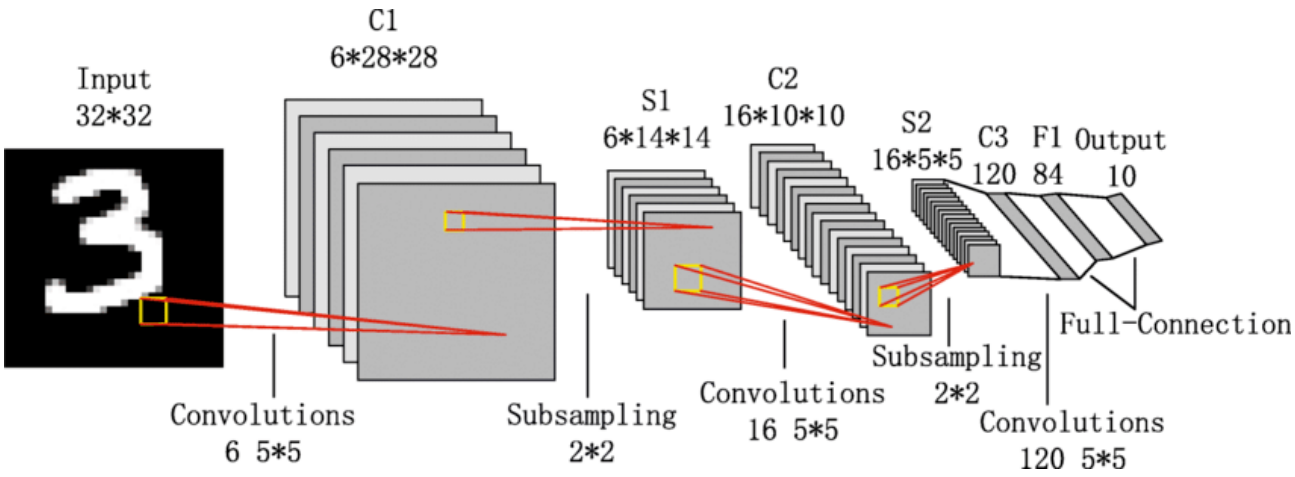

Рисунок 4.1 – Архитектура Lenet5

Преимущества:

1) *Простота и эффективность*
LeNet–5 имеет относительно небольшое количество слоев, что позволяет этой архитектуре эффективно обучаться на небольших наборах данных;

2) *Малые вычислительные ресурсы* LeNet–5 подходит для устройств с ограниченными вычислительными мощностями;

Недостатки:

1) *Не предназначена для сложных задач*
Архитектура плохо справляется с идентификацией объектах на высокоразмерных изображениях из-за малой глубины и отсутствия современных приемов;

2) *Отсутствие инвариантности к масштабам и поворотам* LeNet–5 плохо обрабатывает изображения с объектами, которые сильно изменяют свой размер, ориентацию или расположение в пространстве;

4.2 VGG-16

Архитектура **VGG-16** подходит для более крупных датасетов и достигает точности 71.5% [3]. Входному слою подаются RGB изображения размером 224x224 пикселей. Далее изображения проходят через сверточные слои. Размерность ядер в этих слоях - 3x3. В одной из конфигураций используется сверточный фильтр размера 1x1, который может быть представлен как линейная трансформация входных каналов (с последующей нелинейностью). Сверточный шаг фиксируется на значении 1 пиксель. Пространственное дополнение (padding) входа сверточного слоя выбирается таким образом, чтобы пространственное разрешение сохранялось после свертки, то есть дополнение равно 1 для 3x3 сверточных слоев. Подвыборка осуществляется при помощи пяти max-pooling слоев, которые следуют за одним из сверточных слоев (не все сверточные слои имеют последующие max-pooling). Операция max-pooling выполняется с ядром 2x2 пикселей с шагом 2. После свертки идвут два полносвязных слоя по 4096 нейронов.

1) Входное RGB изображение 224x224.
2) Два сверточных слоя с 64 фильтрами 3x3.
3) Подвыборочный слой Max Pooling 2x2 и шагом 2.
4) Два сверточных слоя с 128 фильтрами 3x3.
5) Подвыборочный слой Max Pooling 2x2 и шагом 2.
6) Два сверточных слоя с 256 фильтрами 3x3.
7) Подвыборочный слой Max Pooling 2x2 и шагом 2.
8) Два сверточных слоя с 256 фильтрами 3x3.
9) Подвыборочный слой Max Pooling 2x2 и шагом 2.
10) Два сета из трех сверточных слоев с 512 фильтрами 3x3.
11) Подвыборочный слой Max Pooling 2x2 и шагом 2.
12) 2 полносвязных слоя с 4096 нейронами.

13) Выходной слой Softmax для классификации на основе признаков лица.

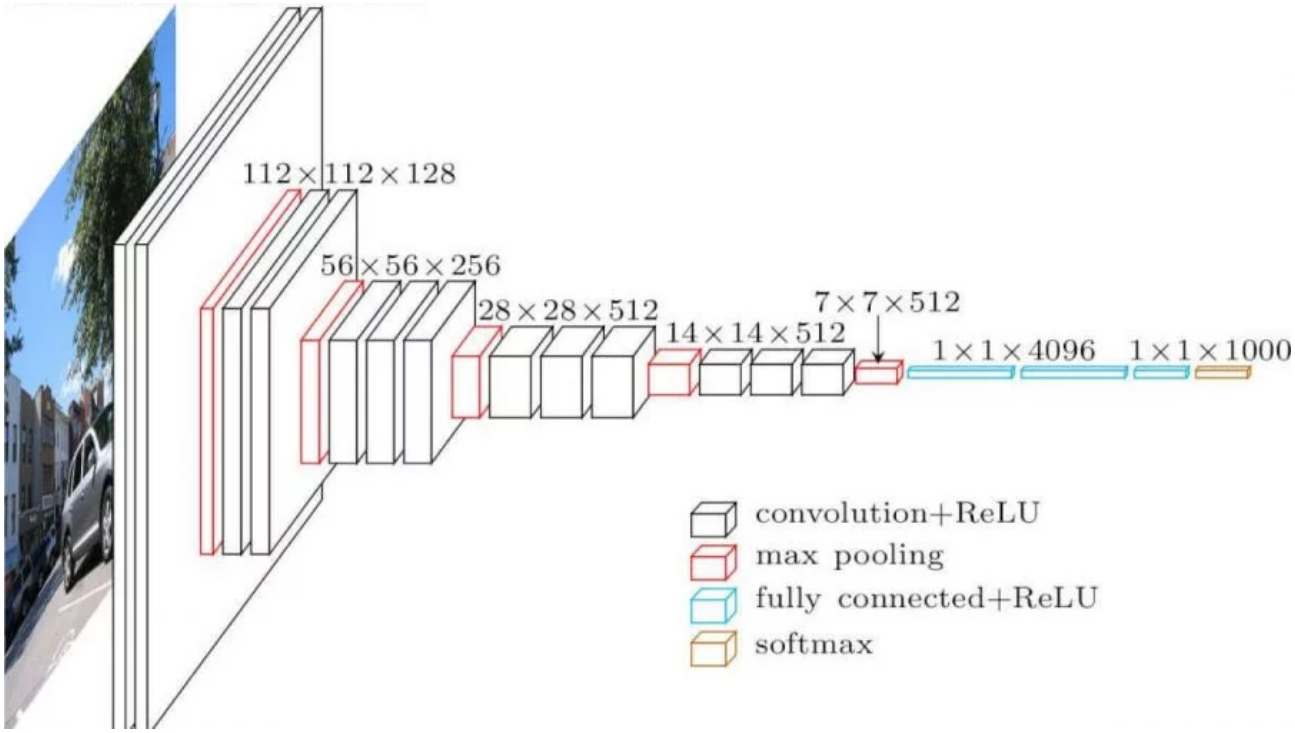

Рисунок 4.2 – Архитектура VGG-16

Преимущества:

1) *Глубокая структура*
   Глубина сети, состоящая из 16 слоев, позволяет эффективно извлекать высокоуровневые признаки из изображений, что способствует высокой точности предсказания на больших датасетах;

2) *Универсальность* VGG-16 может быть использована как для задач классификации, так и для задач детекции объектов.

Недостатки:

1) *Большое время обучения*
   Из-за своей глубины и большого количества параметров сеть требует значительных вычислительных ресурсов для обучения;

2) *Высокая вычислительная сложность*
   Сеть требует мощного GPU и значительного количества времени для обработки изображений.

4.3 YOLO

Архитектура YOLO (you look only once) объединяет в себе и обнаружение, и классификацию объектов за один шаг, в отличие от двухэтапных методов (R-CNN), которые сначала детектируют объект, а потом идентифицируют его [8]. Изображения делится на сетку размеров $S \times S$ (13×13 или 19×19 в зависимости от версии YOLO). Каждая ячейка сетки отвечает за обнаружение объектов, центры которых находятся в пределах этой ячейки. Для каждой ячейки YOLO предсказывает координаты ограничивающего окна и классифицирует изображение, находящееся в этом окне. Схема работы YOLO представлена на рисунке 4.3

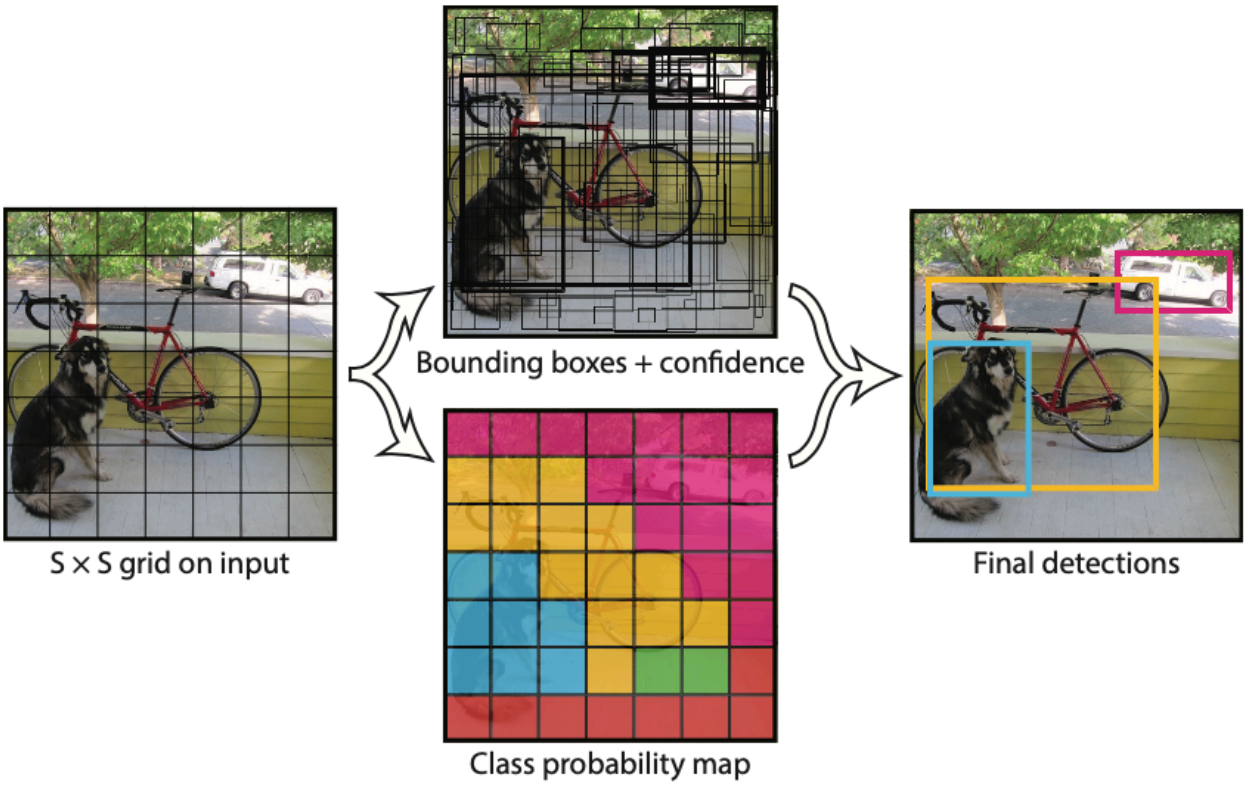

Рисунок 4.3 – Схема работы YOLO

Преимущества:

1) Высокая скорость обработки изображения;

2) Возможность использования в реальном времени.

Недостатки:

1) Плохая работа с маленькими объектами;

2) Точность ниже, чем у некоторых других методов.

4.4 RetinaNet

RetinaNet — это одноэтапный детектор объектов, разработанный для эффективной работы с дисбалансом классов, что делает его подходящим для идентификации компактных объектов, например, на изображениях с космических и летательных аппаратов [2].

Архитектура RetinaNet построена на основе двух основных компонентов: Feature Pyramid Network (FPN) для создания многоуровневого представления объекта и Focal Loss для балансировки обучения. В отличие от YOLO, который работает с сеткой фиксированного размера, RetinaNet использует принцип многоуровневого обнаружения объектов для работы с объектами различных размеров.

Основные этапы работы RetinaNet:

1) Используется база, например ResNet–50 или ResNet–101, для извлечения признаков из изображения. Эти признаки представляют информацию о текстуре, форме и контексте объектов на изображении.

2) На основе извлеченных признаков строится иерархическая структура, где каждый уровень отвечает за объекты определенного размера. Это позволяет алгоритму одинаково хорошо работать с крупными и мелкими объектами.

3) На каждом уровне пирамиды создаются якоря (anchors), представляющие потенциальные объекты разного масштаба и пропорций. Эти якоря служат гипотезами для обнаружения объектов.

4) Для каждого якоря сеть предсказывает:

   - Координаты ограничивающего окна (x, y, w, h),
   - Класс объекта,

- Уверенность в предсказании.

5) Ключевое отличие RetinaNet от других одноэтапных методов — использование Focal Loss. Она снижает влияние легко различимых негативных примеров (фон), сосредотачиваясь на сложных для классификации примерах, таких как маленькие или перекрывающиеся объекты.

Преимущества:

1) Высокая точность идентификации;
2) Эффективная работа с маленькими объектами;
3) Возможность обнаружения объектов различных размеров благодаря FPN.

Недостатки:

1) Низкая скорость работы;
2) Высокая сложность настройки рпараметров для Focal Loss.

Схема работы RetinaNet представлена на рисунке 4.4, где видна работа FPN и использование Focal Loss.

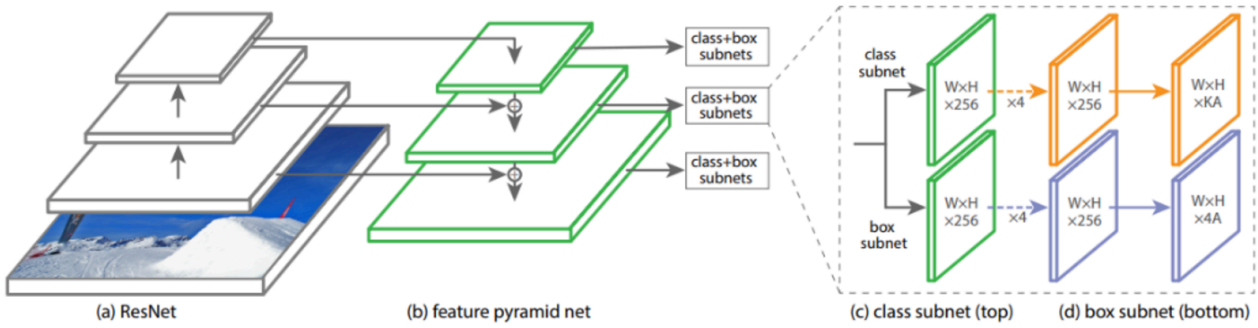

Рисунок 4.4 – Схема работы RetinaNet

4.5 Сравнение

Для сравнения архитектур нейронных сетей для детекции и идентификации объектов на изображениях были выбраны следующие критерии:

1) *Работа в реальном времени* — возможность идентифицировать объекты за промежуток времени, которому соответствует изображение;

2) *Точность* — доля истинно положительных прогнозов из всех прогнозов. Несмотря на то, что это относительная оценка, взятая из разных источников, она может быть сравнена напрямую, так как она измерена на стандартизированных наборах данных (COCO и Pascal VOC).

3) *Количество данных для обучения* — минимальное необходимое количество данных для обучение модели, построенной на основе выбранной архитектуры. Несмотря на то, что это абсолютная оценка, взятая из разных источников, она может быть сравнена напрямую, так как показывает, сколько данных необходимо для обучения модели, чтобы достичь указанной точности, которая измеряется на стандартизированных наборах данных;

4) *Возможность обрабатывать мелкие объекты* — применимость в задачах идентификации мелких объектов на изображениях.

В таблице 4.1 представлено сравнение рассмотренных в этой главе архитектур нейронных сетей для детекции и идентификации объектов на изображениях, где *Архитектура* — название архитектуры нейронной сети, *Realtime* — критерий работы в реальном времени, *Точность* — численный показатель критерия точности, взятый из статей [1, 9], *Данные* — критерий количество данных для обучения, *Мелкие* — возможность обрабатывать мелкие объекты. Символ $+$ обозначает возможность обработки или полную применимость, символ $-$ обозначает невозможность обработки и полную неприменимость, символ $+-$ обозначает, что изначально архитектура не была предназначена для решения такой задачи, но может быть адаптирована под нее.

Таблица 4.1 – Сравнение архитектур нейронных сетей для детектирования и распознавания объектов

Архитектура	*RealTime*	*Точность*	*Данные*	*Мелкие*
LeNet-5	$+$	$\approx 10.2\%$	≈ 60000	$+$
VGG-16	$-$	$\approx 71.5\%$	≈ 1000000	$-$
YOLOv11	$+$	$\approx 81\%$	≈ 100000	$+-$
RetinaNet	$-$	$\approx 70\%$	≈ 100000	$+$

4.6 Вывод

Архитектура YOLO подходит для обработки кадров видеопотоков, так как способна обрабатывать до 150 кадров в секунду, что значительно превыщает частоту смены кадров на современных видеопотоках. При этом нейросеть, построенная на архитектуре YOLO с меньшим успехом справляется с идентификацией мелких объектов, нежели нейросеть, построенная на архитектуре RetinaNet. RetinaNet подходит для идентификации объектов на кадрах, сделанных сделанных с космических и летательных аппаратах.

5 Отслеживание объектов на видеопотоках

В предыдущей главе были рассмотренны алгоритмы, работающие со статическими изображениями. В этой главе будут рассмотренны алгоритмы, работающие с видеопотоками. Видеопоток – это последовательность отдельных кадров, которые были сняты с одной и той же камеры с одним и тем же промежутком времени. Используя данные, полученные с видеопотоков, можно решать задачу отслеживания объектов. Отслеживание объектов – процесс, в ходе которого объекты должны быть ассоциированы в пространстве и времени. При этом при решении задачи отслеживания приходится сталкиваться с проблемой перекрытия. Перекрытие – ситуация, при которой отслеживаемый объект находится за другим непрозрачным объектом или пропадает из кадров текущего видеопотока. В таком случае необходимо предсказывать траекторию движения объекта, чтобы отследить его, после появления объекта на текущем или новом видеопотоке. Таким образом, задачу отслеживание объектов можно представить ввиде диаграммы idef0 на рисунке 5.1.

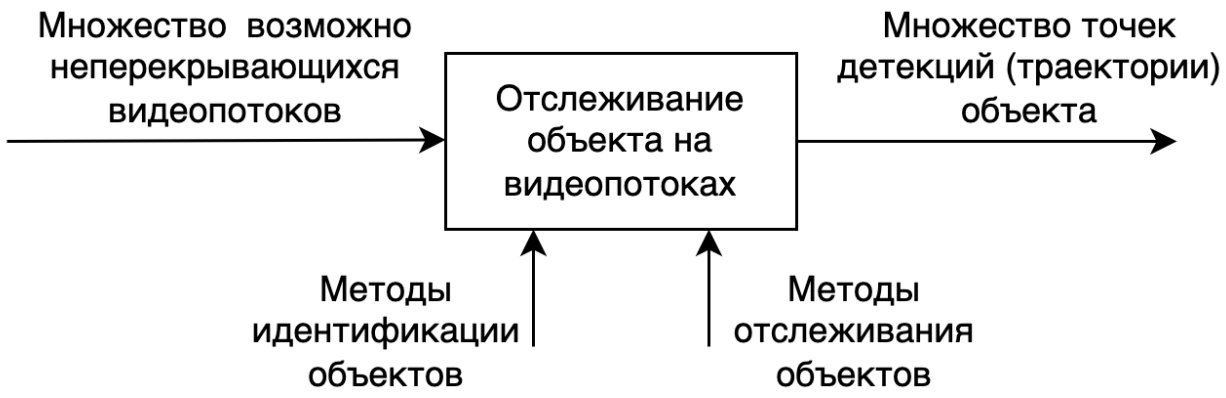

Рисунок 5.1 – Схема алгоритма, решающего задачу отслеживания объекта в формате idef0

5.1 CNN + RNN

Для решения задачи отслеживания объекта на кадрах видеопотока предлагается объединить сверточную нейронную сеть и рекуррентную следующим образом: сверточная нейронная сеть обрабатывает отдельные

кадры видеопотока для извлечения признаков объект, рекуррентная нейронная сеть анализирует последовательность признаков, полученных от сверточной нейронной сети, чтобы учитывать изменение положения объекта во времени. Для каждого кадра x_t используется сверточная нейронная сеть, для извлечения признаков $f_t = CNN(x_t)$. Признаки f_t поступают на вход рекуррентной нейронной сети для вычисления положения объекта $h_t = RNN(h_{t-1}, f_t)$, выходное состояние рекурентной нейронной сети используется для предсказания положения объекта в текущем кадре. Использование рекуррентной сети в данном методе играет важную роль в решении проблемы перекрытия или временной пропажи объекта. Даже если объект временно пропадает из кадра (например, выходит с одной камеры и движется к другой), RNN может использовать информацию о его предыдущих движениях для того, чтобы вычислить возможное местоположение объекта в следующем кадре [10].

Преимущества:

1) *Извлечение пространственных и временных признаков*
   Сверточная сеть эффективно извлекает пространственные признаки объекта, а рекуррентная сеть позволяет учитывать временные зависимости.

2) *Устойчивость к временной потере объекта*
   RNN использует информацию о предыдущих состояниях для предсказания положения объекта, даже если он временно пропадает из кадра.

5.2 SiamFC

Другим решением задачи отслеживания объекта является SiamFC (Siamese Fully Convolutional Network). Сиамская сеть в задаче отслеживания объекта состоит из двух подсетей:

- Подсеть 1 принимает на вход шаблон объекта, который был выделен в предыдущем кадре;
- Подсеть 2 получает текущий кадр (или его часть), где требуется найти

объект.

Эти подсети работают параллельно и извлекают признаки из каждого из входных изображений, а затем их выходы используются для вычисления сходства между шаблоном и текущим фрагментом кадра. Результат этого сходства помогает системе решить, где находится объект в текущем кадре. Когда объект найден, его новый положения используется как обновленный шаблон для последующих кадров. Таким образом, шаблон корректируется по мере перемещения объекта, что позволяет отслеживать объект, даже если он меняет форму. Важно, что когда объект выходит за пределы одного кадра, SiamFC все равно будет искать схожесть между шаблоном и текущим фрагментом изображения. Когда объект появляется в новой области кадра, SiamFC может корректно распознать его и обновить шаблон для отслеживания в следующем кадре [11].

Преимущества:

1) *Устойчивость к деформации объекта*
   Постоянное обновление шаблона позволяет эффективно отслеживать объект, даже если он меняет форму или ориентацию.

2) *Гибкость к типу объектов*
   Постоянное обновление шаблона позволяет эффективно отслеживать объект, даже если он меняет форму или ориентацию.

Недостатки:

1) *Отсутствие обработки временной информации*
   Метод не использует последовательные данные, что ограничивает его эффективность при перекрытии объектов или временной их пропаже.

5.3 DeepSort

Другим решением задачи отслеживания объекта является DeepSORT (Deep Learning for SORT). DeepSORT представляет собой метод отслеживания объектов в реальном времени, который использует рекуррентную нейронную сеть совместно с классической алгоритмической основой SORT (Simple Online and Realtime Tracking).

Алгоритм SORT использует алгоритм Калмана для предсказания положения объектов на основе их предыдущих координат. Он помогает в поддержке стабильных траекторий отслеживаемых объектов в реальном времени [12]. Рекуррентная нейронная сеть используется для извлечения признаков объектов, что улучшает точность идентификации объектов при частичных перекрытиях или изменении их внешнего вида. Сеть обучается на последовательности кадров и помогает улучшить результаты отслеживания, когда объекты выходят из зоны видимости или изменяют своё положение.

Преимущества:

1) *Интеграция Калмана*
   Использование фильтра Калмана позволяет предсказывать движение объекта с высокой точностью, особенно в условиях временной потери данных.

2) *Масштабируемость*
   Подходит для задач с большим количеством объектов в кадре.

Недостатки:

1) *Ограничения при резких изменениях траектории*
   Фильтр Калмана может быть менее точен при резких или нестабильных движениях объекта.

Метод DeepSORT совмещает преимущества алгоритмов Калмана для предсказания местоположения объектов с мощью рекуррентных нейронных сетей для извлечения глубоких признаков, что позволяет улучшить точность отслеживания объектов, несмотря на сложности, такие как перекрытие или частичное скрытие объектов. Алгоритм эффективно решает задачу поддержания уникальной идентификации объектов даже в условиях быстрого движения и изменений в их внешнем виде.

5.4 Сравнение

Для сравнения методов были выбраны следующие критерии:

1) *Возможность обработки в реальном времени* — возможность обрабатывать текущий кадр видеопотока быстрее, чем будет отснят следующий кадр.

2) *Перекрытие* — возможность предсказывания траектории объекта, при его исчезновении.

3) *Изменение объекта* — возможность предсказывать траекторию объекта в ситуации, при которой объект меняет свою форму (разрушается), меняет свое положение или свою ориентацию в пространстве.

4) *Cross–Camera* — возможность адаптировать метод под обработку несколькими камерами.

В таблице 5.1 представлено сравнение методов отслеживания объектов по различным критериям. Для критерия *RealTime* (возможность обработки в реальном времени) символ «+» обозначает возможность обработки видеопотока в реальном времени, не используя больших вычислительных мощностей, символ «−» обозначает невозможность обработки видеопотока в реальном времени, символ «+−» обозначает, что изначально архитектура не была предназначена для обработки видеопотока в реальном времени, но может быть применима, используя большие вычислительные ресурсы. Для критерия *Перекрытие* символ «+» обозначает возможность решения задачи отслеживания объекта при его длительном исчезновении, символ «−» обозначает невозможность решения задачи отслеживания объекта при его коротким исчезновении, символ «+−» обозначает, что изначально архитектура не была предназначена для решения задачи отслеживания траектории объекта при его длительном исчезновении, но может быть применима для решения задачи отслеживания объекта при его коротким исчезновении. Для критерия *Измен. объекта* (изменение объекта) символ «+» обозначает возможность решения задачи отслеживания объекта в ситуации, когда меняется его представление, символ «+−» обозначает возможность решения задачи отслеживания объекта в ситуации, в которой представление объекта меняется незначительно. Для критерия *Cross-Camera* символ «+» обозначает полную применимость архитектуры в системе с системой, состоящей из нескольких камер, символ «+−» обозначает, что архитектура изначально не было

предназначена для работы в системе, состоящей из нескольких камер, символ «−» обозначает полную неприменимость архитектуры в системе, состоящей из нескольких камер.

Таблица 5.1 – Сравнение методов отслеживания объектов по различным критериям

Метод	RealTime	Перекрытие	Измен. объекта	Cross-Camera
CNN+RNN	+−	+	+−	+−
SiamFC	+	−	+	−
DeepSORT	+−	+	+	+

5.5 Вывод

Метод CNN+RNN позволяет учитывать временные изменения положения объекта, а также решать задачи перекрытия и изменения внешнего вида объектов. Кроме того, использование RNN дает значительные преимущества в ситуации временной пропажи объекта или его передвижения между камерами. Однако этот метод может потребовать значительных вычислительных ресурсов для обработки видеопотока в реальном времени, особенно при большом количестве объектов.

Метод SiamFC подходит для задачи отслеживания объектов с высокой точностью и в реальном времени, но имеет ограничения использования в системах, состоящих из нескольких камер. Сиамская сеть подходит для отслеживания объектов внутри одной камеры, однако при переходе объекта между камерами, архитектура сталкивается с трудностями в идентификации объекта, так как шаблон может сильно не соответствовать новому положению объекта. Это делает SiamFC менее подходящим для использования в многокамерных системах или ситуаций с большими изменениями в внешнем виде объекта.

Метод DeepSORT решает задачу отслеживания объектов, используя комбинацию классического алгоритма SORT и рекуррентных нейронных сетей для извлечения признаков. Он работает хорошо в реальном времени и может справляться с частичными скрытиями объектов. Плюсом является способность метода работать в условиях *cross-camera* отслеживания, благодаря

использованию алгоритма Калмана и глубоких признаков, что позволяет поддерживать стабильность траектории объектов, даже если они переходят между камерами. Таким образом, DeepSORT подходит для решения задачи отслеживания в динамичных многокамерных системах.

Выбор метода зависит от специфики задачи. Если требуется отслеживание объектов в реальном времени с учетом временных изменений положения и возможных переходов между камерами, то методы CNN+RNN и DeepSORT будут более предпочтительными. Метод SiamFC лучше всего подойдет для отслеживания объектов внутри одной камеры, где не предполагается значительных изменений в условиях съемки.

ЗАКЛЮЧЕНИЕ

В ходе работы было выяснено, что архитектура YOLO является одним из наиболее эффективных решений для обработки видеопотоков в реальном времени, демонстрируя высокую скорость обработки кадров. Однако она ограничена в идентификации мелких объектов, где более подходящей оказывается архитектура RetinaNet. Последняя проявляет себя лучше в задачах, связанных с анализом сложных изображений, таких как кадры, полученные с космических или летательных аппаратов. Методы, использующие комбинацию CNN и RNN, обеспечивают высокую точность при учете временных изменений положения объекта. Эти подходы хорошо справляются с задачами перекрытия и временной пропажи объекта, что делает их подходящими для многокамерных систем. Однако их применение в реальном времени требует значительных вычислительных ресурсов. Метод SiamFC демонстрирует высокую точность при отслеживании объектов внутри одной камеры, но сталкивается с трудностями при переходе между камерами из-за изменений условий съемки и углов обзора. Метод DeepSORT является универсальным решением для отслеживания объектов в многокамерных системах, эффективно справляясь с перекрытиями, скрытиями объектов и переходами между камерами.

В данной работе была достигнута цель: проведен анализ современных методов идентификации и отслеживания объектов на видеопотоках. Для достижения цели были решены следующие задачи:

1) Представлены современные методы идентификации объектов на изображениях и видео;

2) Проведено сравнение методов идентификации объетов по выбранным критериям;

3) Представлены алгоритмы отслеживания объектов на кадрах видеопотоков;

4) Проведено сравнение методов отслеживания объектов на кадрах видеопотоков.

Результаты проведенного анализа могут быть полезны при разработке систем видеонаблюдения и аналитики, требующих высокой надежности и

точности в условиях динамичной среды.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Comparison of RetinaNet, SSD, and YOLO for real-time pill detection / Yujie Liu, Bo Wu, Zhi Liu [и др.] // BMC Medical Informatics and Decision Making. 2021. Т. 21, № 1. С. 202. URL: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01691-8.

2. Kara Mesut, Ozyurt Gokhan, Yarar Emre. Deep RetinaNet for Dynamic Left Ventricle Detection in Multiview Echocardiography // Computational and Mathematical Methods in Medicine. 2020. Т. 2020. С. 7025403. URL: https://onlinelibrary.wiley.com/doi/10.1155/2020/7025403.

3. Chen Xinyue, Li Wei. Research on VGG16 Convolutional Neural Network Feature Classification Algorithm Based on Transfer Learning // IEEE Xplore. 2022. URL: https://ieeexplore.ieee.org/document/9652277.

4. Мерков А.Б. Распознавание образов. Построение и обучение вероятностных моделей. Москва: ЛЕНАНД, 2014.

5. Калиновский И.А. Методы нейросетевого детектирования лиц в видеопотоке сверхвысокого разрешения. Национальный исследовательский Томский государственный университет, 2016.

6. Haoxiang Li Zhe Lin Xiaohui Shen Jonathan Brand Gang Hua. A Convolutional Neural Network Cascade for Face Detection. Stevens Institute of Technology Hoboken NJ 07030 Adobe Research San Jose CA 95110, 2015.

7. Patrik KAMENCAY Miroslav BENCO Tomas MIZDOS Roman RADIL. A New Method for Face Recognition Using Convolutional Neural Network // ADVANCES IN ELECTRICAL AND ELECTRONIC ENGINEERING. 2017. Т. 15, № 4. С. 663–672.

8. You Only Look Once: Unified, Real-Time Object Detection / Joseph Redmon, Santosh Divvala, Ross Girshick [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016. С. 779–788.

9. Khanam Rahima, Hussain Muhammad. YOLOv11: An Overview of the Key Architectural Enhancements // arXiv preprint arXiv:2410.17725. 2024. URL: https://arxiv.org/abs/2410.17725.

10. Rich feature hierarchies for accurate object detection and semantic segmentation / Ross Girshick, Jeff Donahue, Trevor Darrell [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2014. С. 580–587.

11. Fully-Convolutional Siamese Networks for Object Tracking / Luca Bertinetto, Jack Valmadre, João F. Henriques [и др.] // Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2016. С. 850–865.

12. Authors. Improved DeepSORT-Based Object Tracking in Foggy Weather for Autonomous Vehicles // Sensors. 2023. Т. 24, № 14. С. 4692. URL: https://www.mdpi.com/1424-8220/24/14/4692.

ПРИЛОЖЕНИЕ А

Презентация на 3 слайдах.