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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

«Метод распознавания надводных объектов с аэрофотоснимков с использованием нейронных сетей»

Студент	ИУ7-83Б		Миронов Г. А.
	(Группа)	(Подпись, дата)	(И. О. Фамилия)
Руководитель ВКР			Тассов К. Л.
		(Подпись, дата)	(И. О. Фамилия)
Нормоконтролер			
		(Подпись, дата)	(И. О. Фамилия)

2023 г.

РЕФЕРАТ

Расчетно-пояснительная записка 32 с., 14 рис., 1 табл., 32 источн., 1 прил.

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

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

ANN — (англ. Artificial Neural Network) Искусственная нейронная сеть

CNN — (англ. Convolutional Neural Network) Свёрточная нейронная сеть

RoI — (англ. Region Of Interest) Область интереса

VHR — (англ. Very High Resolution) Сверхвысокое разрешение

MR — (англ. Medium Resolution) Среднее разрешение

ВВЕДЕНИЕ

1 Аналитичекский раздел

1.1 Предметная область

В настояще время все больше внимания привлекает тема автоматизации судоходства, а так же общее повышение уровня безопасмности в процессе эксплуатации судов.

Методы распознавания различных надводных объектов имеют особое значение в данном контексте, так как позволяют решить множество проблем: избежание столкновений судов, осуществление автономного плавания и пр. [1].

Не менее важной является задача распознавания малых надводных объектов, поскольку традиционные методы обнаружения, основанные на использовании радара, не подходят для задачи обнаружения близко расположенных и малых объектов [2].

Кроме того, обнаружение активности рыболовецких судов по-прежнему является сложной задачей для многих стран, расположенных на архипелагах, например — Индонезии. В настоящее время для мониторинга огромной морской акватории используется технология, использующая датчики SAR для обнаружения кораблей, разрабатываемая с 1985 года. Однако стоимость использования данной технологии является одним из основных препятствий для дальнейшего развития [3].

1.2 Нейронные сети

ANN определяется как массово параллельный распределенный процессор, состоящий из простых процессорных блоков, который обладает естественной склонностью накапливать эмпирические знания [4].

Их название и структура вдохновлены человеческим мозгом, а алгоритм работы основывается на способе, которым биологические нейроны передают сигналы друг другу.

В общем случае ANN включает в себя входной слой, выходной (или целевой) слой и, между ними, скрытый слой. Слои соединены через узлы (искусственные нейроны). Эти соединения образуют «сеть» — нейронную сеть — из взаимосвязанных узлов. Пример приведен на рисунке 1.1.

В общем случае искусственный нейрон можно представить в виде регрессионной модели [5], состоящей из входных данных, весовых коэффициентов,

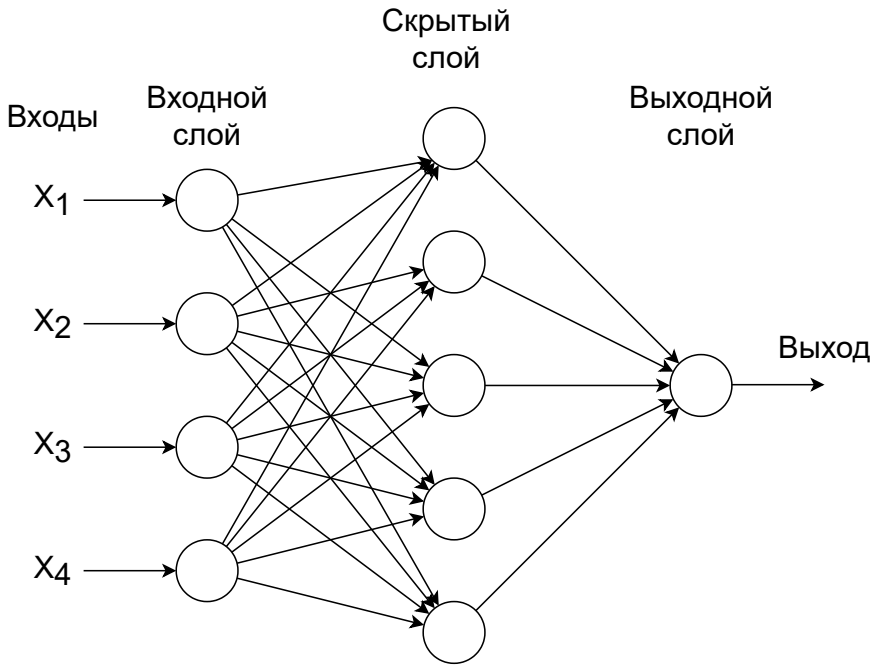

Рисунок 1.1 – Пример схемы ANN

смещения (или порогового значения) и выходных данных. Эту модель можно описать следующей формулой:

$$\hat{y} = \sum_{i=1}^{n} w_i x_i + w_0, \tag{1.1}$$

В качестве функции активации можно использовать: ступенчатую, линейную функции, сигмоиду, ReLu и другие [5].

Модель искусственного нейрона приведенна на рисунке 1.2.

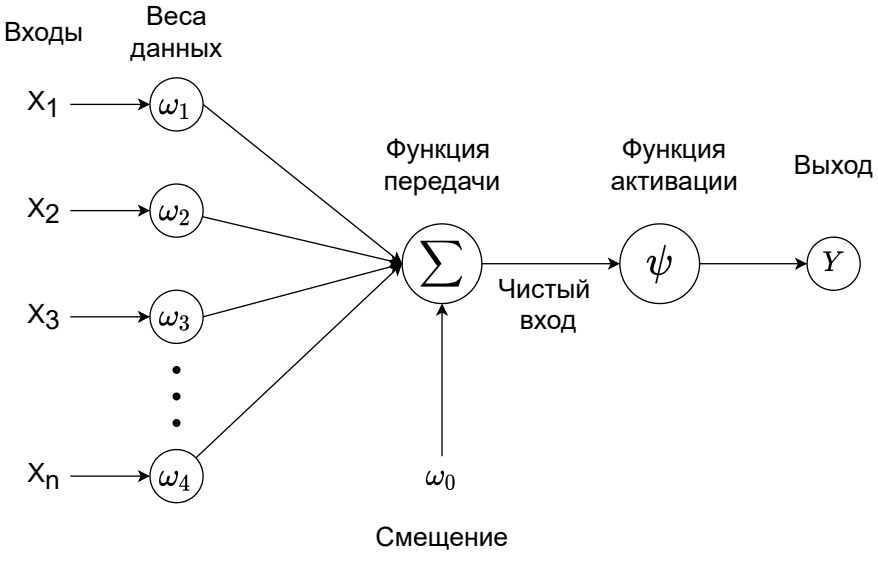

Рисунок 1.2 – Общая схема искусственного нейрона

Данное представление применимо к любому виду нейронных сетей — вне зависимости от типа, нейронные сети реализуются путем упорядочивания нейронов в слои и последующим связыванием соответствующих слоев между собой [4].

В процессе обучения нейронной сети используется так называемая обучающая выборка — заранее подготовленный набор данных, отражающий суть рассматриваемой предметной области [4]. В зависимости от содержимого обучающей выборки результирующие весовые конфигурации нейронной сети (т. е. веса связей между нейронами, а так же смещения отдельно взятых нейронов) могут отличаться [4]. В связи с этим одна и та же структура нейронной сети может переиспользована для работы в различных предметных областях.

1.3 Сверточные нейронные сети

Одним из основных видов нейронных сетей, применяемых для распознавания является CNN [6].

CNN представляет собой тип ANN, которая имеет архитектуру с глубокой обратной связью и выделяется на фоне остальных ANN с полносвязными слоями своей способностью к обобщению. CNN работает с сильно абстрагированными характеристиками объектов, особенно это касается пространственных данных, что позволяет добиться более эффективно идентифицировать объекты в сравнении с другими типами ANN [6]. Одним из отличительных свойств CNN является способность к фильтрации посторонних шумов во входных данных.

Модель CNN состоит из конечного набора уровней обработки, которые могут изучать различные характеристики входных данных (например, изображения) с несколькими уровнями абстракции. Начальные уровни изучают и извлекают высокоуровневые свойства, а более глубокие уровни изучают и извлекают более низкоуровневые свойства. Концептуальная модель CNN представлена на рисунке 1.3.

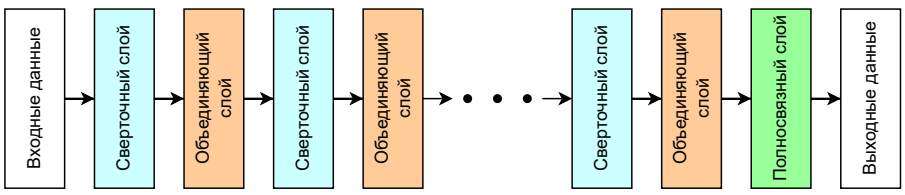

Рисунок 1.3 – Концептуальная модель CNN

Существующие архитектуры CNN для обнаружения объектов на изображениях можно разделить на две категории: одноэтапные (one-stage) и двухэтапные (two-stage) [7].

1.3.1 Свертка

Ядро

Прежде, чем рассматривать процесс свертки, необходимо определить понятие «ядро», используемое при свертке. Ядро предстваляет собой матрицу из дискретных значений или чисел, где каждое значение известно как вес этого ядра. Пример двухмерного ядра приведен на рисунке 1.4.

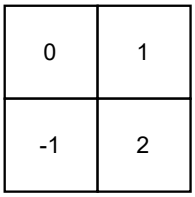

Рисунок 1.4 – Пример двумерного ядра с размерностью 2×2

Ядро инициализируется случайными значениями, которые изменяются в ходе обучения CNN.

Процесс свертки

Свертка — это операция над парой матриц $A(n_x, n_y)$ и $B(m_x, m_y)$, $m_x \leq n_x$, $m_y \leq n_y$, результатом которой является матрица

$$C(n_x - m_x + 1, n_y - m_y + 1) = A * B, \tag{1.2}$$

каждый элемент которой является скалдарным произведением матрицы B (ядра свертки) и некоторой подматрицы A такого же размера.

Т. е. элемент матрицы C вычисляется следующим образом:

$$C_{i,j} = \Sigma_{u=0}^{m_x-1} \Sigma_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v}. \tag{1.3}$$

Пример

Разберем пример свертки для изображения в градациях серого, т.к. такое изображение содержит лишь один канал, передаваемый на вход CNN.

Пусть дано изображение, представленное на рисунке 1.5.

1	0	-2	1
-1	0	1	2
0	2	1	0
1	0	0	1

Рисунок 1.5 – Пример изображения в градациях серого с размерностью 4×4

Далее рассмотрим первые два шага процесса свертки, представленные на рисурках 1.6 и 1.7, соответсвенно.

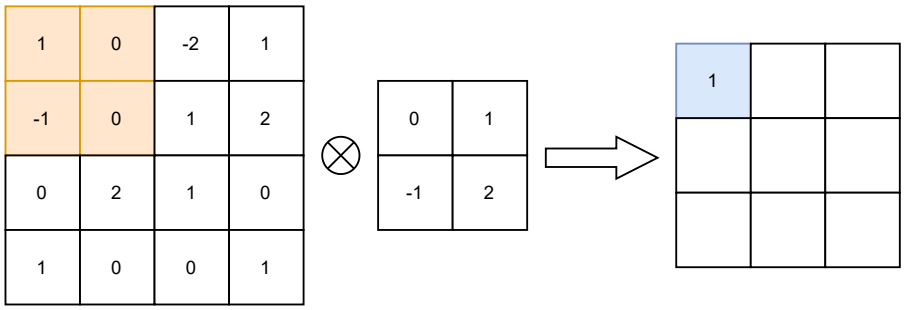

Рисунок 1.6 – Пример свертки изображения в градациях серого с размерностью 4×4

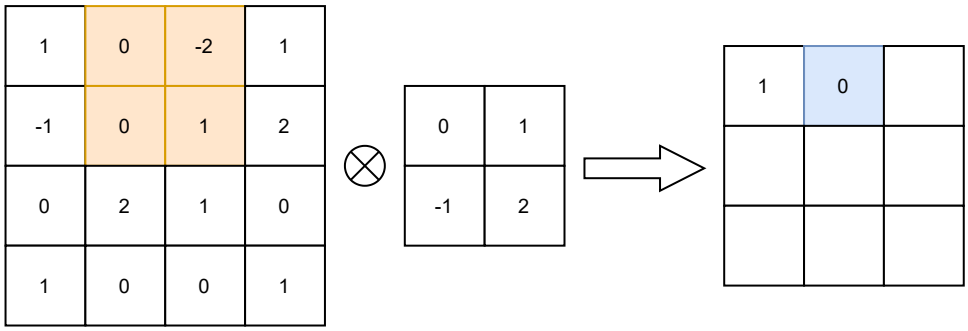

Рисунок 1.7 – Пример свертки изображения в градациях серого с размерностью 4×4

Аналогичным образом свертка продолжается до полного заполнения результирующей матрицы. Стоит отметить, что, в зависимости от размеров окна и перекрытия окон, будет меняться размер результирующей матрицы.

1.3.2 Двухэтапные CNN

В таких нейросетевых алгоритмах выделяют два этапа: поиска RoI (англ. Candidate Region Extraction) на изобаржении и последующей классификации RoI, найденных на первом этапе. При этом под RoI на изображении подразумеваются зоны, потенциально содержащие искомые объекты [8].

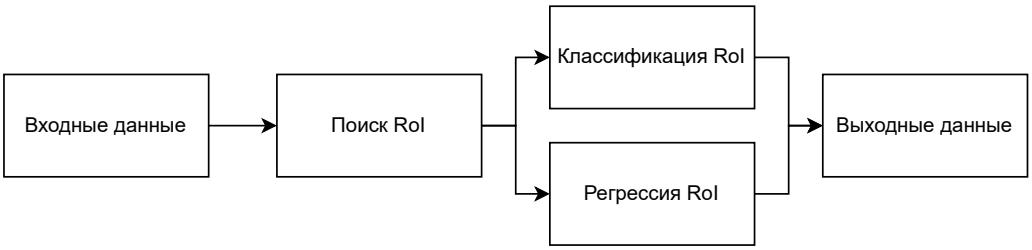

Рисунок 1.8 – Схема работы двухэтапного алгоритма

Стоит отметить, что первый этап может происходить без использования нейронных сетей. Для этого можно использовать информациию о контрасте, ключевые точки или перебор всех возможных положений объекта с помощью процедуры `selective search` [9].

RoI, полученные вышеперечисленными методами, могут обладать серьезными недостатками, например:

— содержать слишком большое количество фона;

— содержать лишь небольшую часть объекта;

— содержать более одного объекта.

В связи с этим на первом этапе более предпочтительным методом является применение CNN, не содержащих полносвязных слоев [9].

На втором этапе CNN применяеются к обнаруженным RoI.

Преимуществом данных алгоритмов является высокая точность распознавания объектов, однако, платой за это является время, необходисмое для выделения «подозрительных» зон на изображении [8].

1.3.3 Одноэтапные CNN

Данные нейросетевые алгоритмы не включают в себя этап поиска RoI на изображении.

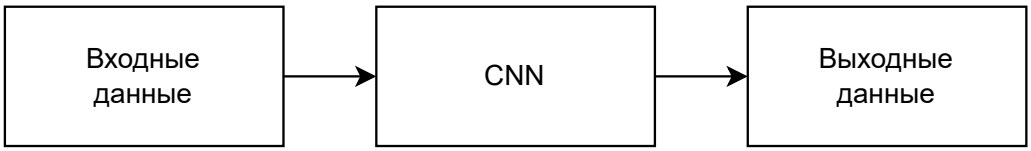

Рисунок 1.9 – Схема работы одноэтапного алгоритма

Преимуществами одноэтапных алгоритмов являются их простота и относительно высокая скорость работы. К недостаткам же можно отнести более низкую точность детектирования объектов по сравнению с двухэтапными алгоритмами, а также меньшую гибкость алгоритма с точки зрения рассматриваемых изображений [7].

1.4 Ансамбли

Ансамбль — это набор слабых экспертов, выполняющих классификацию произвольного объекта $x \in X$, конечный результат которого рассчитывается на основе результатов работы составляющих слабых экспертов. Слабыми экспертами в ансамбле могут выступать как классификаторы, так и другие ансамбли.

Тем не менее, ансамбль на основе нескольких моделей, построенных независимо друг от друга, в общем случае будет иметь более низкое качество, чем ансамбль на основе моделей, построенных с использованием специальных алгоритмов [10].

К этим алгоритмам относятся стекинг, бустинг и бэггинг, позволяющие существенно повысить качество классификации [10].

1.4.1 Стекинг

Стекинг подразумевает параллельные обучение и работу всеч слабых экспертов, т. е. классификаторы не зависят друг от друга [10].

Схема стекинга представлена на рис. 1.10.

Обучающая выборка X разделяется на n случайных равновеликих частей (фолдов).

Для объекта из выборки, находяегося в k-ом фолде, производится предсказание слабыми экспертами, обученными на $k-1$ фолдах. Данный процесс итеративен и происходит для каждого фолда.

Таким обрахом, для каждого объекта обучающей выборки создается

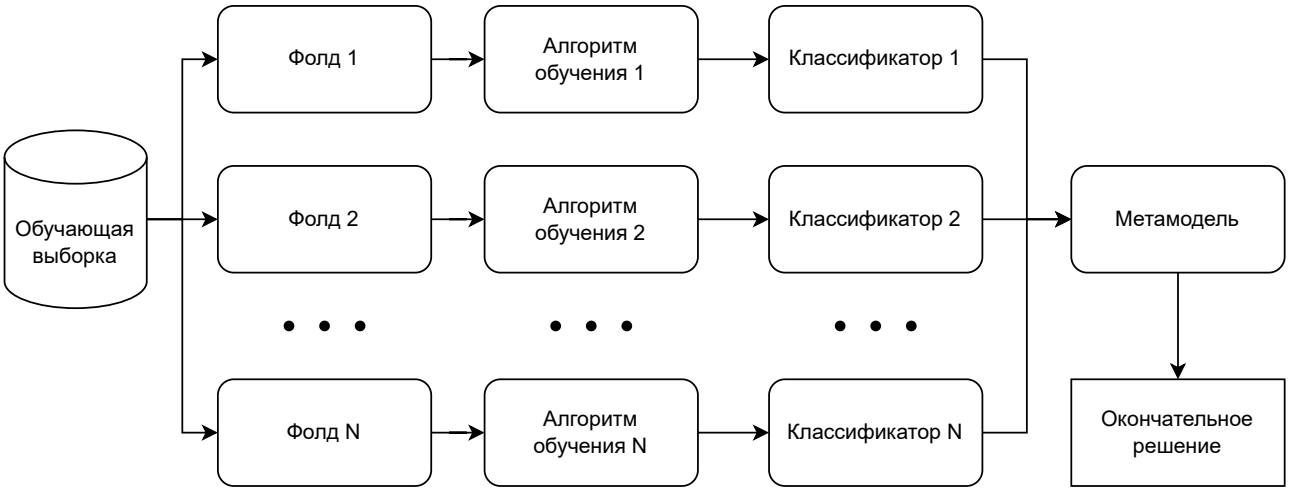

Рисунок 1.10 – Схематическое представление алгоритма стекинга

набор прогнозов слабых экспертов. Далее, на сформированных наборах прогнозов происходит обучение метамодели [10].

1.4.2 Бустинг

При использовании бустинга, каждый последующий алгоритм, входящий в ансамбль, стремится компенсировать недостатки композиции всех предыдущих алгоритмов [10].

Формальная постановка задачи

Пусть $h(a, \overrightarrow{x})$ — слаюый эксперт, где $\overrightarrow{x}$ — это вектор параметров. Необходимо найти следующий алгоритм:

$$H_T(a) = \Sigma_{t=1}^{T} b_t h(a, \overrightarrow{x}), \tag{1.4}$$

где $b_i \in \mathbb{R}$ — коэффициенты, при которых

$$Q = \Sigma_i L(H_T(a_i), y_i) \to min, \tag{1.5}$$

где L — функция потерь.

Так как в общем случае процесс вычисления $\{(\overrightarrow{x}_t, b_t)\}_{t=1}^{T}$ нетривиален, решение находят пошагово:

$$H_t(a) = H_{t-1}(a) + b_t h(a, \overrightarrow{x}). \tag{1.6}$$

Схема бустинга представлена на рис. 1.11.

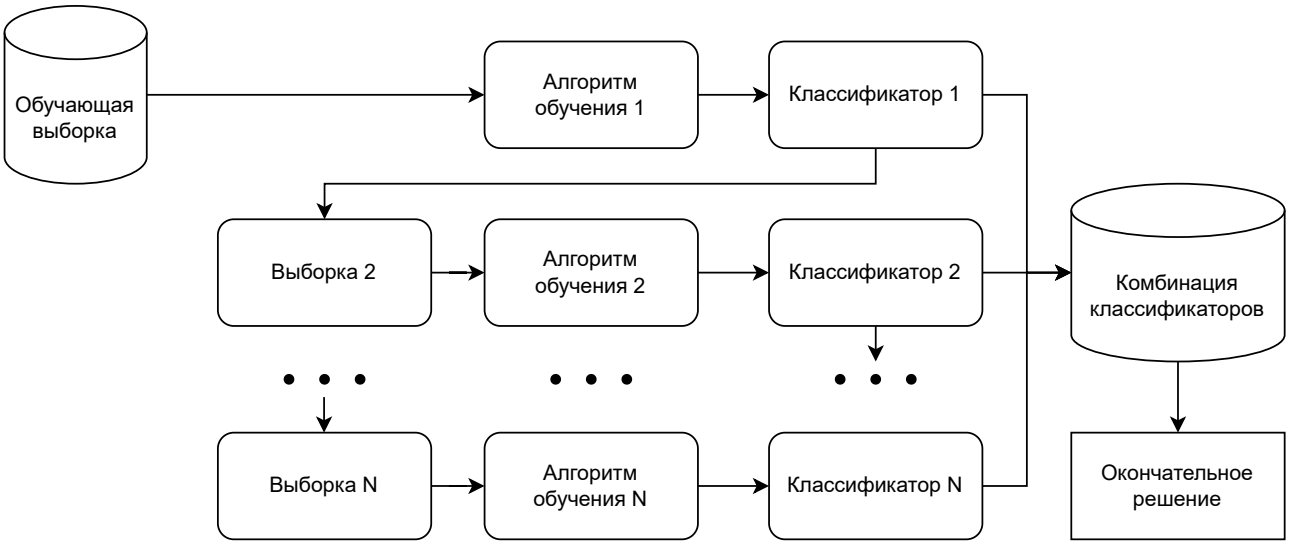

Рисунок 1.11 – Схематическое представление алгоритма бустинга

Алгоритмы

Среди наиболле часто используемых алгоритмов бустинга: AdaBoost и BrownBoost [10].

1.4.3 Бэггинг

В бэггинге все слабые эксперты обучаются и работают параллельно, т. е. независимо друг от друга. При этом обучающая выборка X разделяется на n выборок $X_1, X_2, \ldots, X_n$, причем X_i и X_j могут пересекаться при любых $i, j \in 1 \ldots n$.

Идея данного подхода заключается в том, что в отличие от бустинга классификаторы не исправляют ошибки друг друга, а компенсируют их при голосовании [10].

При этом результат голосования определяется посредством:

— консенсуса — все классификаторы дают одинаковый ответ;

— простого большинства;

— взвешивания — каждому класссификатору присваивается вес, учитываемый при принятии решения.

Схема бэггинга представлена на рис. 1.12.

Преимуществом данного метода перед стекингом является детерминированность результата: мета-модель в стекинге может переобучаться с течением

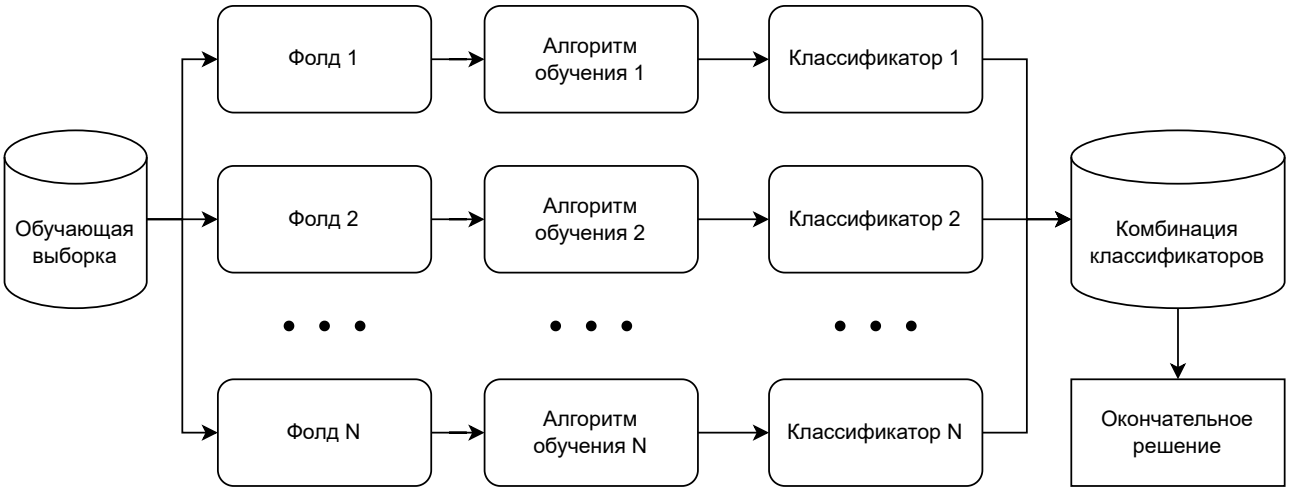

Рисунок 1.12 – Схематическое представление алгоритма бэггинга

времени, в то время как результат голосования в беггинге является детерминированным [10].

1.5 Существующие методы

По результатам проведенного иследования рынка, в настоящий момент не представлены общедоступные системы обнаружения надводных объектов, в связи с чем невозможно определить методы, использующиеся в оных.

Тем не менее, в последние годы тема распознавания надводных объектов является объектом множества исследований, предлагающих разнообразные подходы к решению поставленных задач. Данные методы и будут рассмотрены далее.

1.5.1 Разрешение изображений

В настоящее время существующие методы можно разделить на две группы по разрешению снимков:

— VHR — достигает разрешения в 0.5 метра и менее на 1 пиксель;

— MR — достигает разрешения в несколько метров на 1 пиксель.

Одна из проблем методов, основанных на VHR — доступность данных. В настоящее время изображения с таким разрешением могут быть получены лишь в нескольких источниках, использующих спутниковую съемку.

В тоже время, MR изображения могут быть получены, помимо прочего, с помощью съемки с дронов, что значительно удешевляет процесс сбора данных.

1.5.2 R-CNN

Первой CNN, разработанной для обнаружения объектов, является модель Region-based CNN, которая использует подходы на основе скользящего окна (sliding window) [11].

Здесь авторы разделяют всю задачу на три модуля. В первом модуле из каждого входного изображения извлекаются RoI, которые могут содержать какой-либо объект (с помощью процедуры selective search), затем во втором модуле авторы используют аффинное искажение изображения, чтобы сделать все извлеченные RoI фиксированного размера (или фиксированного соотношения сторон), а затем пропускают эти искаженные RoI через AlexNet CNN для извлечения конечных признаков (векторов признаков фиксированного размера). Наконец, третий этап представляет собой набор линейных SVM (support vector machine), которые причисляют каждый вектор какому-либо классу и отдельный регрессор обрамляющих окон [11].

Последующие версии алгоритма — Fast R-CNN [12] и Faster R-CNN [13] — призваны оптимизировать время работы алгоритма, а так же повысить точность распознавания. Так, в Faster R-CNN процедура selective search была заменена на сеть предложений регионов (RPN). RPN — это CNN, используемая для создания высококачественных RoI.

CNN данного семейства активно используются в задачах, допускающих повышение времени работы системы ради повышения точности распознавания, т.е. в задачах, не относящихся к системам реального времени.

Faster R-CNN

Данная CNN является последним опубликованным улучшением алгоритма R-CNN. Основное отличие от предшествующей Fast CNN является замена процедуры генерации претендентов избирательным поиском на нейронную сеть, которая использует имеющуюся карту особенностей [13].

Итоговый принцип работы Faster R–CNN представлен на рисунке

Внесенные изменения позволяют повысить быстродействие распознавания образов на изображении вплоть до десяти раз в сравнении с предшествующей версией алгоритма [13].

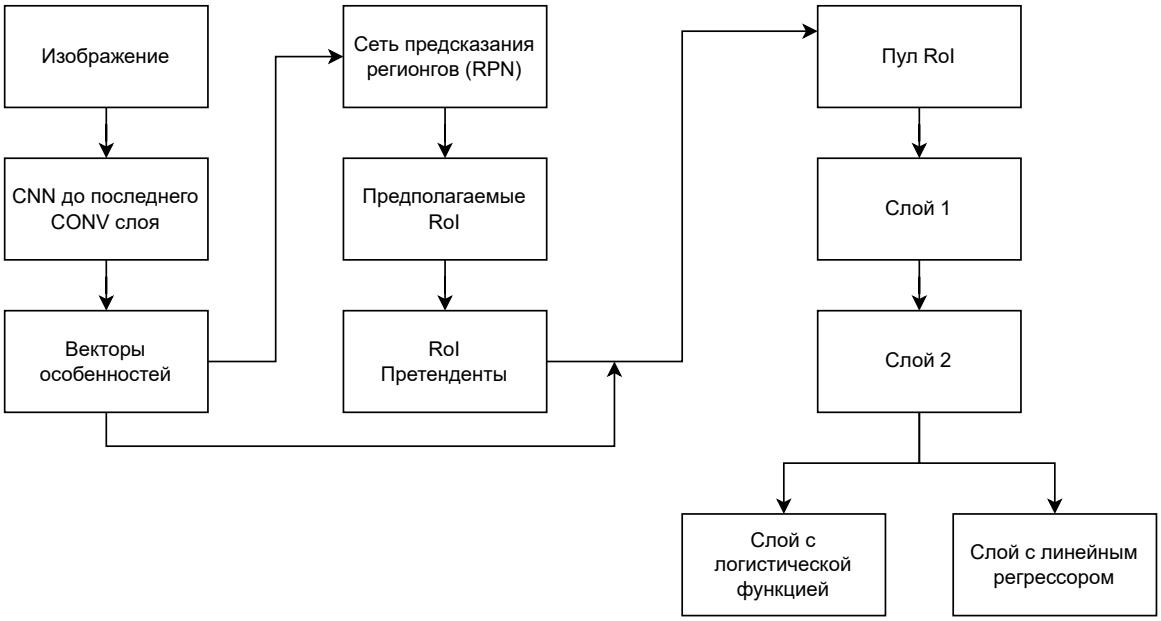

Рисунок 1.13 – Принцип работы Faster R–CNN

1.5.3 YOLO

Алгоритм YOLO (You Only Look Once) является одноэтапным и может непосредственно распознавать объекты, а также их местоположение с помощью сквозной обученной модели CNN [14].

В исходном алгоритме YOLO входное изображение разбивается на фиксированное число сеток, а затем из каждой сетки предсказывается фиксированное число местоположений обрамляющих окон (bounding boxes) и вероятностей. Затем используется пороговое значение для выбора и определения местоположения объекта на изображении [14].

Основной проблемой YOLO является более низкая точность при распознавании больших и малых объектов, а так же присущая всем одноэтапным алгоритмам потеря точности распознавания в сравнении с двухэтапными алгоритмами [14].

Данное семейство алгоритмов включает в себя множество оптимизаций оригинального YOLO, причем последней опубликованной является YOLOv8 [15], представленная в 2023 году.

Начиная с YOLOv4, варианты алгоритма имеют малозначительные изменения, призванные улучшить его характеристики в контексте конкретных задач [16]. Например, одним из вариантов развития актуальной на тот момент времени YOLOv3 [17] является YOLOv3 tiny [18], разработанная для решения

задачи распознавания судов и кораблей в режиме реального времени.

Данное семейство CNN активно используется в задачах распознавания надводных объектов. В первую очередь, это связано с малыми затратами времени на обработку изображения, в сравнении с другими алгоритмами. Так же, достоинством YOLO является, возможность распознавать большое число объектов на одном изображении [19].

YOLOv5

В настоящее время данная CNN, разработанная Ultralytics в 2020 году, пользуется большой популярностью. Данная версия была опубликована вскоре после выхода YOLOv4, однако основным ее отличем стало использование PyTorch [20] вместо Darknet [21] в качестве средства реализации. Поддержка и развитие осуществляется за счет сообщества, в связи с чем до сих пор не опубликована научная работа, описывающая данную CNN [15].

В момент написания данной работы, актуальной является версия v7.0. Кроме того, существуют следующие версии масштабирования CNN:

— YOLOv5n (nano);

— YOLOv5s (small);

— YOLOv5m (medium);

— YOLOv5l (large);

— YOLOv5x (extra large).

Отдельно стоит отметить, что перечисленные выше версии данной CNN имеют так же варианты с различным числом выходных слоев для объектов — от P3 до P6, соответственно, и обозначаемые, например YOLOv5s6 для YOLOv5s с P6.

В зависимости от версии масштабирования изменяется точность и время обработки единичного изображения, соответственно [15].

YOLOv8

Данная CNN так же разработана Ultralytics. В ней применены несколько новых подходов как к процессу обучения, так и к самой архитектуре сети.

Аналогично YOLOv5, существует 5 масштабных версий:

— YOLOv8n (nano);

— YOLOv8s (small);

— YOLOv8m (medium);

— YOLOv8l (large);

— YOLOv8x (extra large).

Смысл именования сохранен.

К сожалению, из-за новизны данной архитектуры, еще не были опубликованы результаты сравнения точности работы и производительности в сравнении с предыдующими версиями и, в частности, с YOLOv5.

Исходя из представленной разработчиком информации, данная CNN имеет более высокую точность обнаружения объектов, однако уступает прелылушей версии в быстродействии.

1.5.4 Параметры для сравнения

Для оценки работы нейронной сети используется величина AP (Average Precision), вычисляемая следующим образом:

$$AP = \frac{\text{кол-во верно распознанных объектов}}{\text{общее кол-во распознанных объектов}}. \tag{1.7}$$

Отметим, что AP вычисляется для объектов одного класса, в связи с чем для классификаторов с несколькими возможными классами объектов используется величина mAP (англ. mean AP) — среднее значение AP.

Кроме того, данную величину принято оценивать в зависимости от IoU (Intersection over Union), вычисляемую по формуле:

$$IoU = \frac{\text{площадь пересечения областей}}{\text{площадь объединения областей}}. \tag{1.8}$$

IoU описывает то, насколько предсказанные CNN обрамляющие окна близки к «истинным». Данная величина принимает значения в диапазоне $[0; 1]$, соответственно.

1.5.5 Сравнение рассмотренных методов

YOLOv5x

На основании датасета MS COCO test-dev 2017, достигается значение AP равное 50.7% при размере входного изображения в 640 пикслелей. При увеличении размера изображения до 1536 пикселей AP достигает 55.8% [15].

Кроме того, использование батча с размером 32, достигается производительность вплоть до 200 FPS при использовании NVIDIA V100 [22].

YOLOv8x

Заявляется, что при использовании того же датасета YOLOv8x достигает AP равного 53.9% при размере входного изображения в 640 пикслелей (в сравнении с 50.7% для YOLOv5x) [15].

Производительность данной CNN достигает 280 FPS при использовании NVIDIA A100 [23] и TensorRT [24].

Сравнение методов

Приведенные в таблице результаты получены при размере входного изображения равном 640×640 пикселей. Измерения проводились на основании датасета COCO [25] с использованием GPU NVIDIA GeForce RTX 4090 [26].

Таблица 1.1 – Сравнение рассмотренных методов

CNN	AP_{IoU}		Параметры, млн. шт.	FLOPs, млрд.	FPS
	$AP_{0.5}$	$AP_{0.5:0.95}$			
Faster R-CNN	62.5	—	53	888	—
YOLOv5n	45.7	28.0	1.9	4.5	934
YOLOv5x	50.7	68.9	86.7	205.7	252
YOLOv8n	37.3	50.4	3.2	8.7	1163
YOLOv8x	53.9	—	68.2	257.8	236

1.6 Метод распознавания надводных объектов с аэрофотоснимков с использованием нейронных сетей

1.7 Формализованная постановка задачи

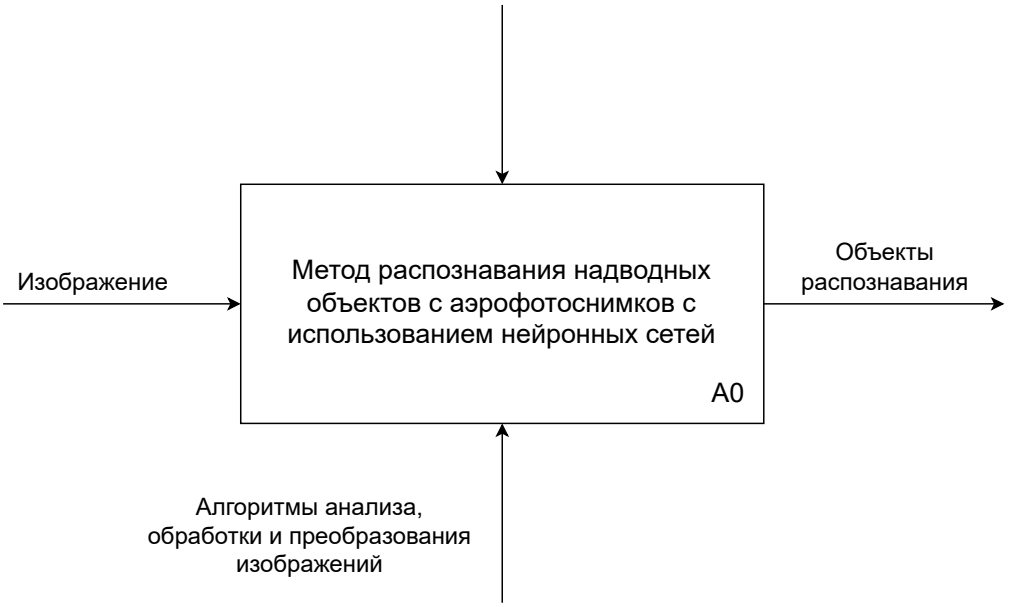

Рисунок 1.14 – Общий вид метода

1.8 Выбор данных для обучения модели

[27]

2 Конструкторский раздел

2.1 Требования к разрабатываемому методу

2.2 Требования к разрабатываемому программному комплексу

2.3 Проектирование метода обнаружения

2.4 Структура разрабатываемого программного комплекса

2.5 Данные для обучения модели

В качестве данных для обучения моделей были выбраны два набора данных:

— kaggle-ships-in-google-earth [28];

— VAIS_RGB+SMD+MARITIME+WSODD+MARVEL [29].

3 Технологический раздел

3.1 Средства реализации

3.1.1 Выбор языка программирования

Для написания программного комплекса будет использоваться язык программирования Python 3 [30]. Данный выбор обусловлен следующими факторами:

— широкий набор библиотек для работы с нейронными сетями;

— возможность тренировать нейронную сеть на графическом процессоре с использованием технологии CUDA [31].

3.1.2 Выбор библиотеки глубокого обучения

Для создания и обучения модели нейронной сети была выбрана библиотека PyTorch [20] версии 2.0.0. Выбор данной версии обусловлен поддержкой CUDA 11.8, предоставляемой GPU NVIDIA GeForce RTX 2060 [32], на котором будет производиться обучение нейронной сети.

3.2 Реализация программного комплекса

3.2.1 Тренировка модели

3.3 Результаты обучения модели

3.4 Примеры использования разработанного программного комплекса

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Lee S.-J.*, *Roh M.-I.*, *Oh M.-j.* Image-based ship detection using deep learning // Ocean Systems Engineering. — 2020. — Т. 10. — С. 415—434. — DOI: 10.12989/ose.2020.10.4.415.
2. *Z. Chen* Deep learning for autonomous ship-oriented small ship detection / Z. Chen [и др.] // Safety Science. — 2020. — Т. 130. — С. 104812.
3. *M. Marzuki* Fishing boat detection using Sentinel-1 validated with VIIRS Data / M. Marzuki [и др.] // IOP Conference Series: Earth and Environmental Science. — 2021. — Т. 925. — С. 012058. — DOI: 10.1088/1755-1315/925/1/012058.
4. *Sharkawy A.-.-N.* Principle of Neural Network and Its Main Types: Review // Journal of Advances in Applied & Computational Mathematics. — 2020. — Т. 7. — С. 8—19. — DOI: 10.15377/2409-5761.2020.07.2.
5. *Sharma S.*, *Sharma S.*, *Athaiya A.* ACTIVATION FUNCTIONS IN NEURAL NETWORKS // *S. Sharma* International Journal of Engineering Applied Sciences and Technology. — 2020. — Т. 4. — С. 310—316. — DOI: 10.33564/IJEAST.2020.v04i12.054.
6. *A. Ghosh* Fundamental Concepts of Convolutional Neural Network / A. Ghosh [и др.]. — 2020. — DOI: 10.1007/978-3-030-32644-9_36.
7. *Zhang H.*, *Cloutier R.* Review on One-Stage Object Detection Based on Deep Learning // EAI Endorsed Transactions on e-Learning. — 2022. — Т. 7. — С. 174—181. — DOI: 10.4108/eai.9-6-2022.174181.
8. *Du L.*, *Zhang R.*, *Wang X.* Overview of two-stage object detection algorithms // Journal of Physics: Conference Series. — 2020. — Т. 1544. — С. 012033. — DOI: 10.1088/1742-6596/1544/1/012033.
9. *А. В. Бондаренко* АЛГОРИТМ НЕЙРОСЕТЕВОГО РАСПОЗНАВАНИЯ НАДВОДНЫХ ОБЪЕКТОВ В РЕАЛЬНОМ ВРЕМЕНИ / А. В. Бондаренко [и др.] // Известия Тульского государственного университета. Технические науки. — 2021. — Т. 1. — С. 19—33.
10. *M. Ganaie* Ensemble deep learning: A review / M. Ganaie [и др.]. — 2021. — DOI: 10.48550/arXiv.2104.02395.

11. *R. Girshick* Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation / R. Girshick [и др.] // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — 2013. — DOI: 10.1109/CVPR.2014.81.

12. *Girshick R.* Fast R-CNN. — 2015. — DOI: 10.1109/ICCV.2015.169.

13. *S. Ren* Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / S. Ren [и др.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2015. — Т. 39. — DOI: 10.1109/TPAMI.2016.2577031.

14. *P. Jiang* A Review of Yolo Algorithm Developments / P. Jiang [и др.] // Procedia Computer Science. — 2022. — Т. 199. — С. 1066—1073. — ISSN 1877-0509. — DOI: https://doi.org/10.1016/j.procs.2022.01.135.

15. *Terven J.*, *Cordova-Esparza D.-M.* A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. — 2023. — Апр.

16. *L. Hao* Enhanced YOLO v3 Tiny Network for Real-time Ship Detection from Visual Image / L. Hao [и др.] // IEEE Access. — 2021. — Т. PP. — С. 1—1. — DOI: 10.1109/ACCESS.2021.3053956.

17. *Redmon J.*, *Farhadi A.* YOLOv3: An Incremental Improvement. — 2018.

18. *D. Li* Yolo-tiny-MSC: A tiny neural network for object detection / D. Li [и др.] // Journal of Physics: Conference Series. — 2021. — Т. 1873. — С. 012073. — DOI: 10.1088/1742-6596/1873/1/012073.

19. *Redmon J.*, *Farhadi A.* YOLO9000: Better, Faster, Stronger // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — С. 6517—6525. — DOI: 10.1109/CVPR.2017.690.

20. PyTorch [Электронный ресурс]. — Режим доступа: https://pytorch.org/ (Дата обращения: 22.04.2023).

21. Darknet [Электронный ресурс]. — Режим доступа: https://github.com/pjreddie/darknet (Дата обращения: 22.04.2023).

22. NVIDIA V100 TENSOR CORE GPU [Электронный ресурс]. — Режим доступа: https://www.nvidia.com/en-us/data-center/v100/ (Дата обращения: 22.04.2023).

23. NVIDIA A100 Tensor Core GPU [Электронный ресурс]. — Режим доступа: https://www.nvidia.com/en-us/data-center/a100/ (Дата обращения: 22.04.2023).

24. NVIDIA TensorRT [Электронный ресурс]. — Режим доступа: https://developer.nvidia.com/tensorrt (Дата обращения: 22.04.2023).

25. COCO - Common Objects in Context [Электронный ресурс]. — Режим доступа: https://cocodataset.org/ (Дата обращения: 20.11.2022).

26. GeForce RTX 4090 [Электронный ресурс]. — Режим доступа: https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4090/ (Дата обращения: 23.04.2023).

27. Kaggle. Airbus Ship Detection Challenge [Электронный ресурс]. — Режим доступа: https://www.kaggle.com/c/airbus-ship-detection (Дата обращения: 30.03.2023).

28. kaggle-ships-in-google-earth Dataset [Электронный ресурс]. — Режим доступа: https://universe.roboflow.com/robin-public/kaggle-ships-in-google-earth-dfqwt (Дата обращения: 19.04.2023).

29. VAIS_RGB+SMD+MARITIME+WSODD+MARVEL Dataset [Электронный ресурс]. — Режим доступа: https://universe.roboflow.com/wilson_xu_weixuan-outlook-com/vais_rgb-smd-maritime-wsodd-marvel (Дата обращения: 19.04.2023).

30. Welcome to Python.org [Электронный ресурс]. — Режим доступа: https://www.python.org/ (Дата обращения: 23.04.2023).

31. What is CUDA | NVIDIA official blog [Электронный ресурс]. — Режим доступа: https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/ (Дата обращения: 23.04.2023).

32. Introducing The GeForce RTX 2060: Turing For Every Gamer [Электронный ресурс]. — Режим доступа: https://www.nvidia.com/en-us/geforce/news/gfecnt/nvidia-geforce-rtx-2060/ (Дата обращения: 23.04.2023).

ПРИЛОЖЕНИЕ А