

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Основная часть</b>	<b>5</b>
1.1 Формальная постановка задачи . . . . .	5
1.2 Бэггинг . . . . .	6
1.3 Объединение результатов работы слабых экспертов . . . . .	7
1.4 Структура разрабатываемого программного комплекса . . . . .	8
1.5 Данные для обучения модели . . . . .	9
1.6 Обогащение данных . . . . .	9
1.7 Обучение и тестирование слабых экспертов . . . . .	10
1.8 Средства реализации . . . . .	10
1.9 Реализация программного комплекса . . . . .	11
1.9.1 Модуль метода распознавания . . . . .	11
1.9.2 Обучение слабых экспертов . . . . .	11
1.10 Результаты обучения слабых экспертов . . . . .	12
1.11 Примеры использования разработанного программного комплекса	13
<b>ЗАКЛЮЧЕНИЕ</b>	<b>14</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>15</b>
<b>ПРИЛОЖЕНИЕ А Схемы алгоритмов</b>	<b>16</b>

## ВВЕДЕНИЕ

Во время выполнения выпускной квалификационной работы был разработан метод распознавания надводных объектов с аэрофотоснимков.

Целью настоящей работы является разработка и реализация программного обеспечения для проверки осуществимости концепции разработанного метода.

# 1 Основная часть

## 1.1 Формальная постановка задачи

Входными данными метода является изображение. Результатом работы являются объекты распознавания, а именно — список обрамляющих окон, соответствующих распознанным объектам.

На рисунке 1.1 представлена диаграмма, описывающая общий вид метода распознавания надводных объектов с использованием нейронных сетей.

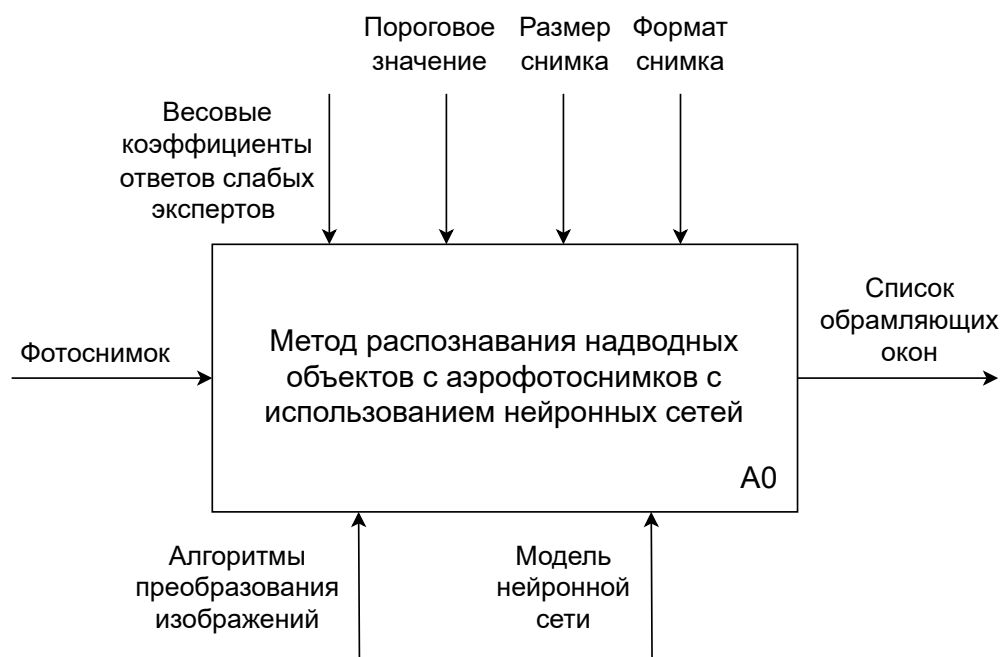
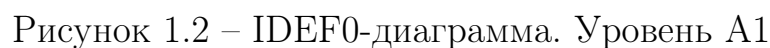


Рисунок 1.1 – Общий вид метода

На метод накладываются следующие ограничения:

- входные данные — фотоснимок в формате PNG, JPG, JPEG;
- размерность входного изображения — не ниже  $640 \times 640$  и не более  $1280 \times 1280$  пикселей;
- разрешение входного изображения — от 0.5 до 15 метров/пиксель;
- работа только в дневное время суток (так как метод, разработанный для «ночной» работы может обладать совершенно другими свойствами);
- распознавание только различных объектов на изображении.

На рисунке 1.2 представлена IDEF0 диаграмма метода уровня A1.



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

При этом в данной работе результат голосования определяется посредством взвешивания — каждому классификатору присваивается вес, учитываемый при принятии решения.

Преимуществом данного метода перед стекингом является детерминированность результата: мета-модель в стекинге может переобучаться с течением времени, в то время как результат голосования в беггинге является детерминированным [1].

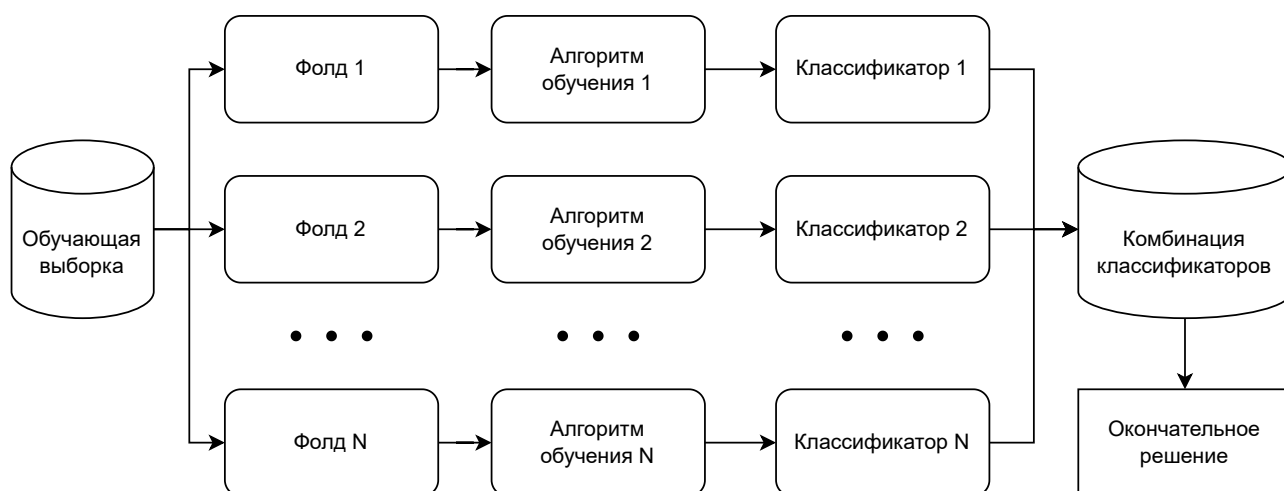


Рисунок 1.3 – Схематическое представление алгоритма бэггинга

### 1.3 Объединение результатов работы слабых экспертов

Результатом работы слабых экспертов является множество обрамляющих окон, а так же соответствующие меры достоверности результатов.

В общем случае, обрамляющие окна, полученные в результате работы слабых экспертов могут пересекаться или же полностью дублироваться. Кроме того, в разрабатываемом методе предполагается учитывать весовых коэффициентов соответствующих слабых экспертов для корректировки результатов работы в случае, когда контекст снимка заранее известен.

Соответствующая схема алгоритма представлена на рисунке А.1.

Для того, чтобы обнаружить пересечение обрамляющих окон, необходимо установить порог значения IoU, при превышении которого окна считаются пересекающимися. Такое решение обусловлено контекстом решаемой задачи — при съемке с большого расстояния наводные объекты могут частично накладываться, что должно быть учтено в разрабатываемом методе.

Схема алгоритма обнаружения пересечений обрамляющих окон представлена на рисунке А.2

В процессе обработки результатов работы слабых экспертов возможна ситуация, при которой пересекающиеся окна имеют одинаковый вес и, соответственно, становится невозможным выбор лишь одного из окон.

В качестве алгоритма разрешения подобных конфликтов предлагается использовать выбор наиболее близкого к центру масс пересечения какого-либо из подмножеств пересекающихся окон. При этом под наиболее близким

понимается окно с наименьшим Евклидовым расстоянием между центром окна и центром масс, соответственно.

Соответствующая схема алгоритма представлена на рисунке А.3

## 1.4 Структура разрабатываемого программного комплекса

Разрабатываемый программный комплекс состоит из двух модулей:

- модуль, реализующий метод распознавания объектов с использованием модели YOLOv8;
- пользовательское приложение, производящее распознавание объектов на основе разработанного метода, а так же предоставляющее возможность обучения модели, используемой в предыдущем модуле.

### Модуль YOLOv8 модели

В данном модуле происходит только обучение и сохранение модели. Модуль должен быть использован либо при первоначальном обучении модели, либо при последующем дообучении в связи с вносимыми изменениями.

Результатом работы модуля является файл, содержит в себе обученную модель, т.е. ее структуру и весовые коэффициенты соответствующих связей. Это необходимо для того, чтобы не тратить время на обучение модели при повторном использовании, так как процесс обучения может занимать продолжительный период времени в зависимости от некоторых факторов.

На рисунке 1.4 представлена схема работы с модулем YOLOv8 модели.

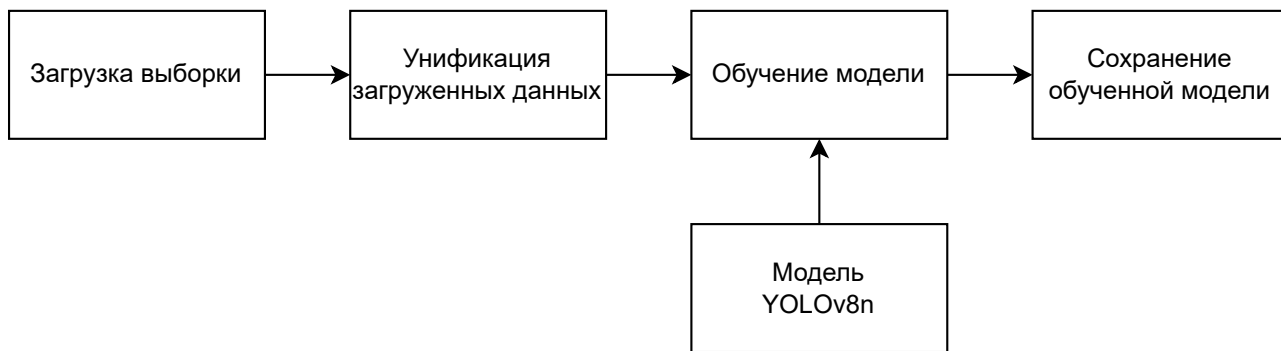


Рисунок 1.4 – Схема работы с модулем YOLOv8 модели

## Модуль пользовательского приложения

Данный модуль позволяет пользователю загружать изображение для последующего анализа и выбора параметров настройки метода (весовых коэффициентов слабых экспертов). Результатом работы модуля является изображение с выделенными надводными объектами.

На рисунке 1.5 представлена схема работы с модулем пользовательского приложения.

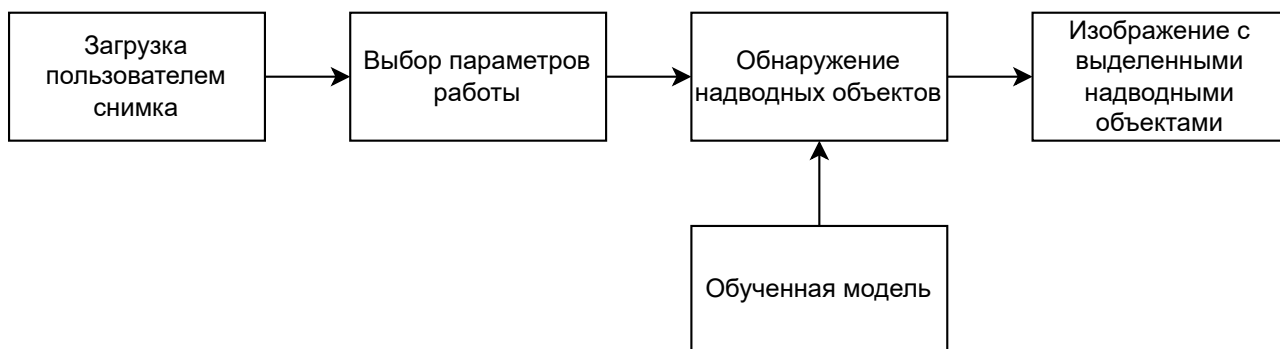


Рисунок 1.5 – Схема работы с модулем пользовательского приложения

### 1.5 Данные для обучения модели

В качестве данных для обучения моделей были выбраны три общедоступных набора данных: Kaggle Ships in Google Earth [2], ShipRSImageNet [3] и Huawei Ship [4].

В таблице 1.1 приводится общая информация о каждом из используемых наборов данных.

Таблица 1.1 – Сравнение наборов данных

Набор данных	Число		Ракурс
	снимков, шт.	объектов, шт.	
Kaggle Ships	1658	> 2000	Спутник
ShipRSImageNet	3435	> 17500	Спутник
Huawei Ship	5538	8132	БПЛА

### 1.6 Обогащение данных

В связи с малым числом изображений в обучающей выборке Huawei Ship, для каждого из представленных снимков по отдельности применены были

применены трансформации (с различными параметрами): горизонтальное отражение, поворот (до 15 градусов), различные алгоритмы размытия, выравнивание гистограммы, исключение (до 20%), наложение Гауссова шума, изменение цветового тона и резкости.

## **1.7 Обучение и тестирование слабых экспертов**

Так как в данной методе используется бэггинг, то каждый из слабых экспертов будет обучаться независимо от остальных. В связи с этим, выборка должна быть разделена на  $n$  непересекающихся выборок, где  $n$  — число слабых экспертов.

В общем случае для обучения и тестирования слабых экспертов полученные после разделения выборки примерно разделяются в соотношении 80 : 15 : 5 на обучающую, тестовую и валидационную подвыборки, соответственно.

С учетом приведенного выше описания наборов данных, в методе распознавания будет использовано 3 слабых эксперта, обучаемых на соответствующем наборе данных. Тестирование каждого из обученных слабых экспертов будет производится на основании данных, представленных в соответствующем наборе данных, в связи со спецификой каждого из последних.

## **1.8 Средства реализации**

### **Выбор языка программирования**

Для реализации программного комплекса будет использоваться язык программирования Python 3 [5]. Данный выбор обусловлен следующими факторами:

- широкий набор библиотек для работы с нейронными сетями;
- возможность обучать нейронную сеть на графическом процессоре с использованием технологии CUDA [6].

Кроме того, как указывалось ранее, для выбранного языка программирования уже существуют общедоступные пакеты для работы с выбранным методом обнаружения, включая методы для обучения, тестирования и валидации работы метода.



## **Выбор библиотеки глубокого обучения**

Для создания и обучения модели нейронной сети была выбрана библиотека PyTorch [7] версии 2.0.0. Выбор данной версии обусловлен поддержкой CUDA 11.8, предоставляемой GPU NVIDIA GeForce RTX 2060 [8], на котором будет производиться обучение нейронной сети.

### **1.9 Реализация программного комплекса**

#### **1.9.1 Модуль метода распознавания**

Реализация модуля построена на YOLOv8n модели. Модуль состоит из трех слабых экспертов, обученных с использованием различных наборов данных, как было описано ранее.

В связи с тем, что данные слабые эксперты не зависят от результатов работы друг друга, присутствует возможность распараллеливания процесса обработки изображения слабыми экспертами с целью снижения времени отклика системы.

Для организации параллельной обработки будет использован Ray — фреймворк с открытым исходным кодом, предоставляющий возможность параллельного выполнения, а так же кластеризации различного рода вычислений, в том числе при использовании нейронных сетей [9].

#### **1.9.2 Обучение слабых экспертов**

В предоставляемом Ultralytics пакете для работы с YOLOv8 предусмотрены методы для обучения моделей с различными параметрами и возможностью применения таких оптимизаций, как батчинг изображений, мозаичная аугментация на ранних стадиях обучения, отслеживание результатов процесса обучения с прерыванием в случае простоя и прочие [10].

Помимо этого, указанный выше пакет предоставляет средства для автоматической предобработки изображения (его масштабирования) при превышении размера, использованного при обучении модели. В совокупности с использованием нормализованных координат при формации результатов распознавания, это позволяет полностью переложить процесс предобработки на описанный выше пакет.

## 1.10 Результаты обучения слабых экспертов

В таблице 1.2 приведена общая информация по итогам обучения слабых экспертов в зависимости от набора данных.

Таблица 1.2 – Сравнение результатов обучения на наборах данных

Набор данных	Всего эпох, шт.	Порог		Достигнутая точность, %
		эпоха, шт.	точность, %.	
Kaggle Ships	100	30	80	92
ShipRSImageNet	100	40	60	85
Huawei Ship	15	—	—	83

На рисунке 1.6 приведена графическая интерпретация процесса обучения слабых экспертов.

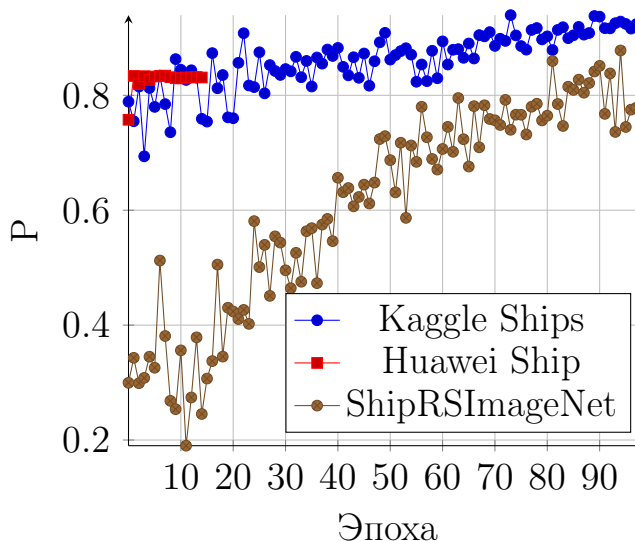


Рисунок 1.6 – Точность слабых экспертов (P)

Различия в кривых обучения между Kaggle Ship и ShipRSImageNet объясняются отличиями в содержимом наборов данных, а так же различиями в точности разметки данных в различных наборах.

В процессе обучения с использованием Huawei Ship, модель завершила обучение на 15ой эпохе. Данный факт был установлен экспериментально — в течение 50-ти эпох после этого не было отмечено значительных изменений в точности обнаружения объектов.

## 1.11 Примеры использования разработанного программного комплекса

Модуль, реализующий метод распознавания объектов с использованием модели YOLOv8 выполнен в виде пакета на языке Python 3 и не предоставляет пользовательского интерфейса для работы с ним.

В то же время, модуль пользовательского приложения представляет собой консольное приложение без графического интерфейса, конфигурируемые при помощи аргументов командной строки.

В листингах 1.1 и 1.2 приведены примеры взаимодействия с пользовательским модулем.

Листинг 1.1 – Взаимодействие с пользовательским приложением обучения модели

```
$ python3 src/train.py --help
usage: train.py [-h] [-d DATA] [-e EPOCHS]

options:
  -h, --help            show this help message and exit
  -d DATA, --data DATA  Путь к набору данных
  -e EPOCHS, --epochs EPOCHS
                        Число эпох для обучения
```

Листинг 1.2 – Взаимодействие с пользовательским приложением распознавания объектов

```
$ python3 cli.py --help
usage: cli.py [-h] [-m MODEL] [-i IMAGE] [-o OUTPUT]

options:
  -h, --help            show this help message and exit
  -m MODEL, --model MODEL
                        Путь к файлу/директории с обученной моделью
  -i IMAGE, --image IMAGE
                        Путь к файлу снимка
  -o OUTPUT, --output OUTPUT
                        Директория для сохранения результатов
```

## **ЗАКЛЮЧЕНИЕ**

Было разработано и реализовано программное обеспечение, демонстрирующее практическую осуществимость разработанного в ходе выполнения выпускной квалификационной работы метода распознавания надводных объектов с аэрофотоснимков.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *M. Ganaie* Ensemble deep learning: A review / M. Ganaie [и др.]. — 2021. — DOI: 10.48550/arXiv.2104.02395.
2. 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).
3. *Z. Zhang* ShipRSImageNet: A Large-scale Fine-Grained Dataset for Ship Detection in High-Resolution Optical Remote Sensing Images / Z. Zhang [и др.] // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. — 2021. — DOI: 10.1109/JSTARS.2021.3104230.
4. Huawei\_ship Dataset [Электронный ресурс]. — Режим доступа: [https://universe.roboflow.com/huaweiship/huawei\\_ship](https://universe.roboflow.com/huaweiship/huawei_ship) (Дата обращения: 07.05.2023).
5. Welcome to Python.org [Электронный ресурс]. — Режим доступа: <https://www.python.org/> (Дата обращения: 23.04.2023).
6. 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).
7. PyTorch [Электронный ресурс]. — Режим доступа: <https://pytorch.org/> (Дата обращения: 22.04.2023).
8. 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).
9. Productionizing and scaling Python ML workloads simply | Ray [Электронный ресурс]. — Режим доступа: <https://www.ray.io/> (Дата обращения: 16.05.2023).
10. ultralytics - PyPI [Электронный ресурс]. — Режим доступа: <https://pypi.org/project/ultralytics/> (Дата обращения: 09.05.2023).

# ПРИЛОЖЕНИЕ А

## Схемы алгоритмов

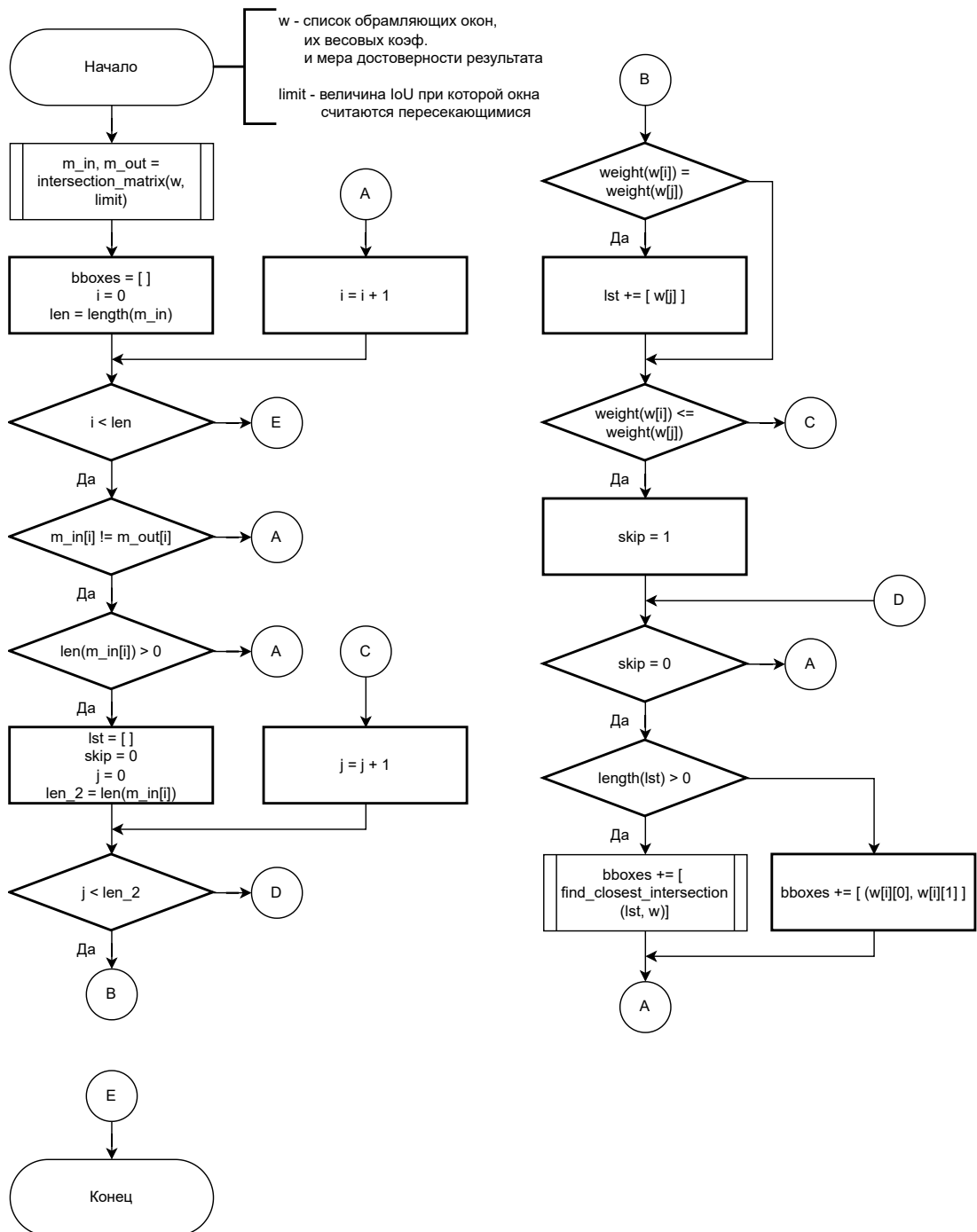


Рисунок А.1 – Схема алгоритма объединения результатов работы слабых экспертов

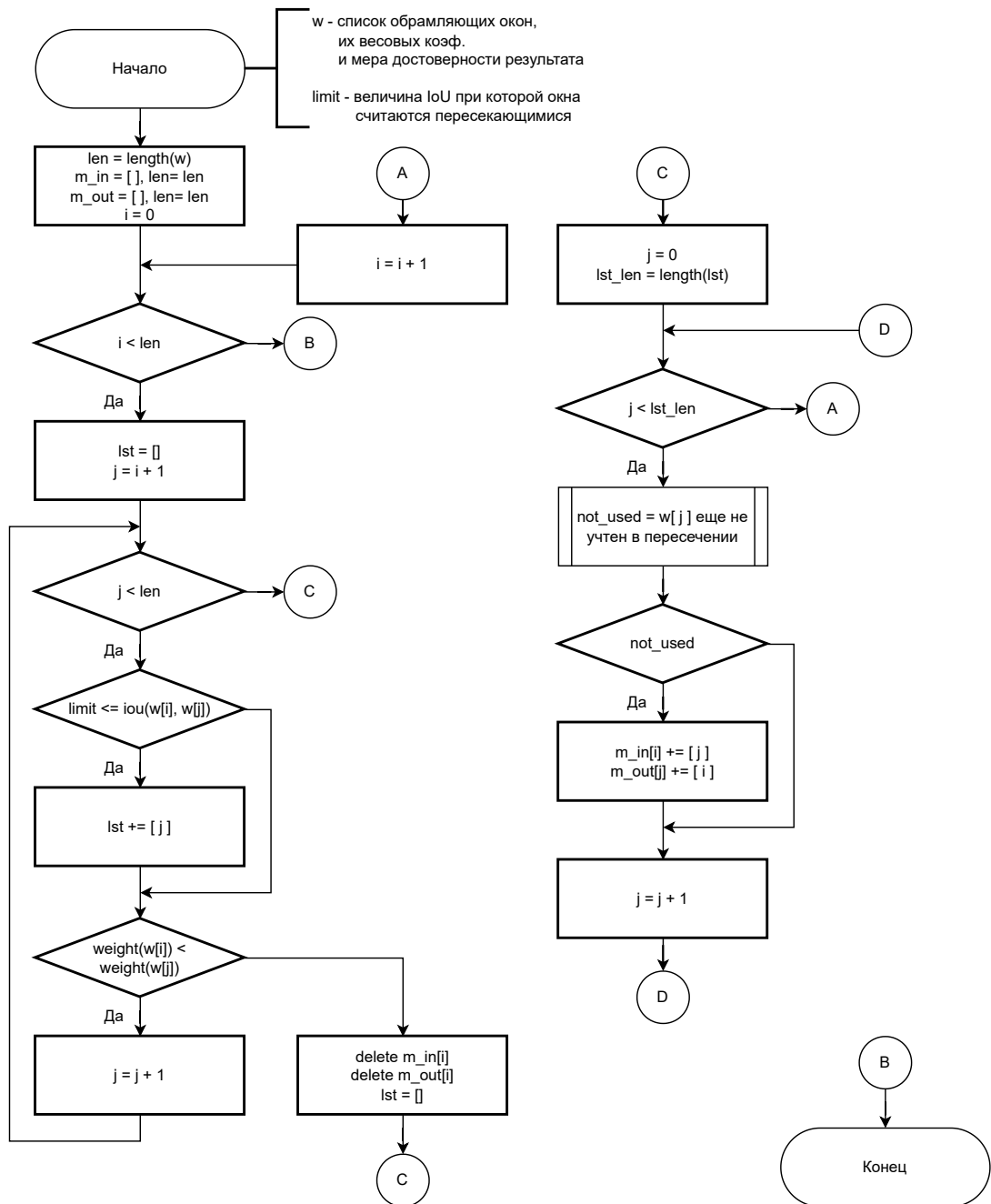


Рисунок А.2 – Схема алгоритма обнаружения пересечений обрамляющих окон

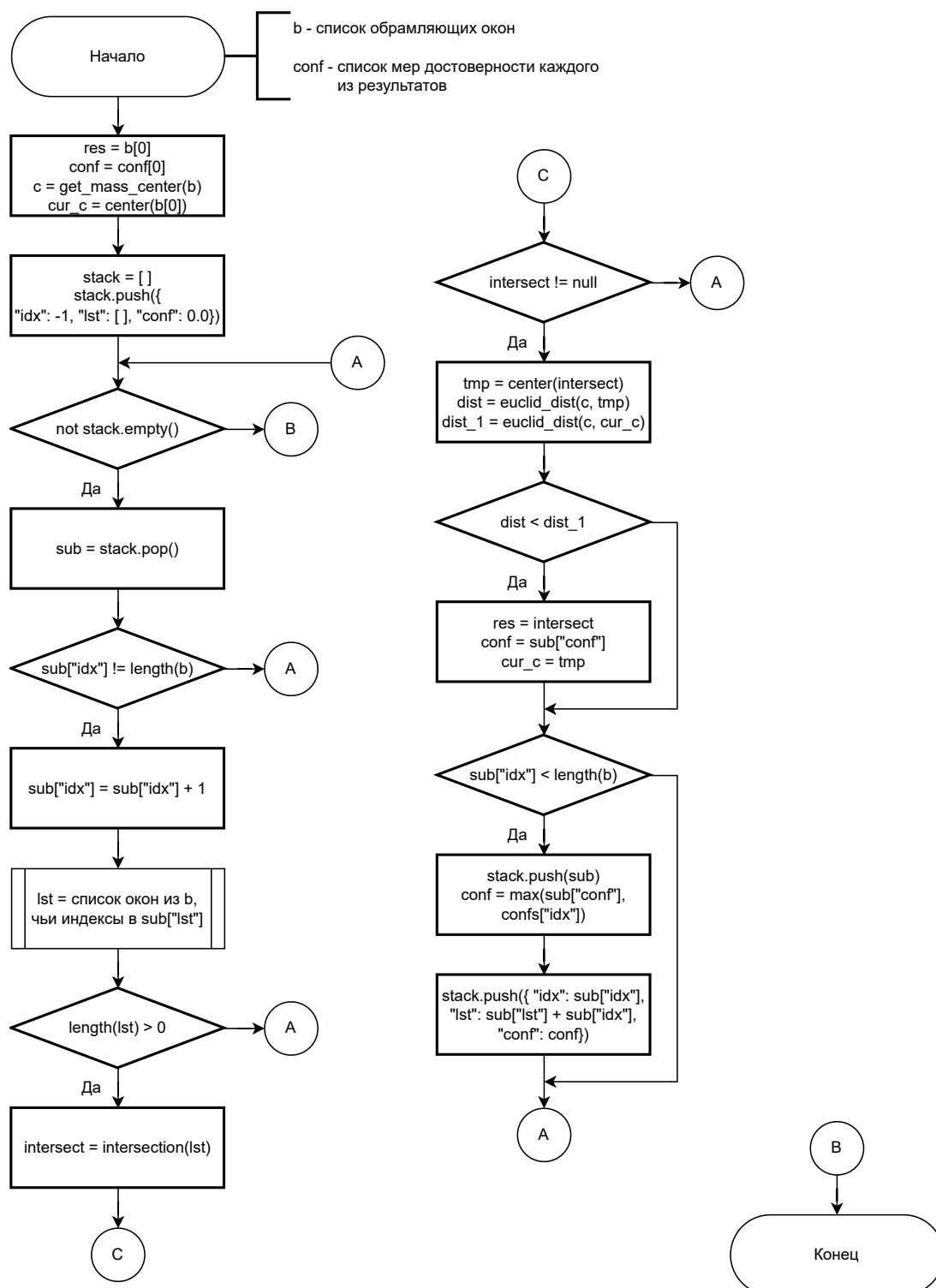


Рисунок А.3 – Схема алгоритма пересечения обрамляющих окон