



Рисунок 2 — Результат работы программы

**Заключение.** Реализация интеллектуальной системы, способной распознавать диатомовые водоросли — это решение технически и математически сложной задачи, требующей углубленного изучения предметной области, собственного анализа, умения и навыков при обработке и выборе экспериментальных данных, глубокие знания в области дискретной математики, программирования, а также биологии и многих других областях науки.

Разработанное приложение позволит классифицировать диатомовые водоросли и тем самым сократить время в определении качества водных ресурсов.

#### Список цитируемых источников

1. Семейкин, В. Д. Моделирование искусственных нейронных сетей в среде MATLAB // В. Д. Семейкин, А. В. Скупченко. — Сист. телекоммуникаций, 2008.
2. Шапович, Е. Г. Автоматическая классификация диатомовых водорослей / Е. Г. Шапович. // Современные технологии и автоматизация в технике, управлении и образовании : сб. тр. I Междунар. науч.-практ. конф., 20 дек. 2018 г. — М. : НИЯУ МИФИ ; Балаково : БИТИ НИЯУ МИФИ, 2019. — С. 202—206

УДК 004.932.72'1

**Е. Г. Шапович**

*Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь*

## ОБНАРУЖЕНИЕ ОБЪЕКТОВ С ПОМОЩЬЮ ГЛУБОКОГО ОБУЧЕНИЯ

**Введение.** В связи с тем, что обнаружение объектов тесно связано с анализом видео и пониманием изображений, в последние годы оно привлекло большое внимание исследователей. Традиционные методы обнаружения объектов основаны на функциях ручной работы и неглубокой обучаемой архитектуре. Их производительность легко снижается из-за создания сложных совокупностей, которые объединяют несколько низкоуровневых функций изображения с высокоуровневым контекстом из детекторов объектов и классификаторов сцен. С быстрым развитием глубокого обучения появляются более мощные инструменты, которые могут изучать семантические, высокоуровневые и более глубокие функции для решения проблем, существующих в традиционных архитектурах. Эти модели ведут себя по-разному в сетевой архитектуре, стратегии обучения, функции оптимизации и т. д.

**Основная часть.** Чтобы получить полное представление об изображении, мы должны не только сосредоточиться на классификации различных изображений, но также попытаться точно оценить концепции и расположение объектов, содержащихся в каждом изображении. Эта задача называется обнаружением

объекта, которое обычно состоит из различных подзадач, таких как обнаружение лиц, обнаружение пешеходов и обнаружение множества точек (скелета).

Как одна из фундаментальных проблем компьютерного зрения, обнаружение объектов способно предоставить ценную информацию для семантического понимания изображений и видео и связано со многими приложениями, включая классификацию изображений, анализ поведения человека, распознавание лиц и беспилотное вождение. Между тем, унаследовав от нейронных сетей и связанных систем обучения, прогресс в этих областях приведет к разработке алгоритмов нейронных сетей, а также окажет большое влияние на методы обнаружения объектов, которые можно рассматривать как системы обучения. Однако из-за больших различий в точках обзора, позе, окклюзии и условиях освещения трудно идеально выполнить обнаружение объекта с помощью дополнительной задачи локализации объекта. В последние годы к этой области было привлечено очень много внимания.

Определение проблемы обнаружения объектов состоит в том, чтобы определить, где объекты расположены на данном изображении (локализация объекта) и к какой категории принадлежит каждый объект (классификация объектов). Таким образом, конвейер традиционных моделей обнаружения объектов можно в основном разделить на три этапа: выбор информативной области, выделение признаков и классификация.

Поскольку разные объекты могут появляться в любых положениях изображения и иметь разные соотношения сторон или размеры, естественным выбором является сканирование всего изображения с помощью многомасштабного скольжения. Хотя эта исчерпывающая стратегия может выявить все возможные положения объектов, ее недостатки также очевидны. Из-за большого количества окон-кандидатов — это требует больших вычислительных ресурсов и создает слишком много избыточных окон. Однако, если применяется только фиксированное количество шаблонов скользящего окна, могут быть получены неудовлетворительные области.

Чтобы распознать различные объекты, нам нужно выделить визуальные особенности, которые могут обеспечить семантическое и надежное представление. Характерными особенностями являются функции масштабно-инвариантной трансформации признаков (SIFT — scale-invariant feature transform), гистограммы направленных градиентов (HOG — Histogram of Oriented Gradients) и признаки цифрового изображения, Хаара (признаки цифрового изображения, используемые в распознавании образов). Это связано с тем, что эти особенности могут создавать представления, связанные со сложными клетками человеческого мозга. Однако из-за разнообразия внешнего вида, условий освещения и фона сложно вручную создать надежный дескриптор объекта, который бы идеально описывал все виды объектов.

Кроме того, необходим классификатор, чтобы отличить целевой объект от всех других категорий и сделать представления более иерархичными, семантическими и информативными для визуального распознавания. Обычно хорошим выбором является метод опорных векторов (SVM — support vector machine) [1], AdaBoost [1] и модель на основе деформируемых деталей (DPM) [1]. Среди этих классификаторов DPM представляет собой гибкую модель, объединяющую части объекта со стоимостью деформации, чтобы справиться с серьезными деформациями. В DPM с помощью графической модели сочетаются тщательно разработанные низкоуровневые функции и кинематическая декомпозиция деталей. Дискриминационное изучение графических моделей позволяет создавать высокоточные модели на основе деталей для различных классов объектов.

Однако, на основе этих дискриминантных дескрипторов локальных функций и неглубоких обучаемых архитектур можно получить незначительное количество успешных методов [1]. Этот факт обусловлен следующими причинами:

1. Создание ограничивающих рамок-кандидатов с помощью стратегии скользящего окна является избыточным, неэффективным и неточным.
2. Семантический разрыв не может быть преодолен с помощью комбинации дескрипторов низкого уровня, созданных вручную, и неглубоких моделей с дискриминативным обучением.

Благодаря появлению глубоких нейронных сетей (DNN), более значительный выигрыш был получен с введением Regions с функциями сверточных нейронных сетей CNN (R-CNN — Region Based Convolutional Neural Networks). DNN или наиболее представительные CNN действуют совершенно иначе, чем традиционные подходы. У них более глубокая архитектура, позволяющая изучать более сложные функции, чем мелкие. Кроме того, выразительность и надежные алгоритмы обучения позволяют изучать информативные представления объектов без необходимости проектирования элементов вручную [1].

С момента предложения R-CNN было предложено множество улучшенных моделей, в том числе Fast R-CNN, который совместно оптимизирует задачи классификации и регрессии ограничивающего прямоугольника, Faster R-CNN, который требует дополнительной подсети и YOLO, который выполняет обнаружение объектов с помощью регрессии с фиксированной сеткой. Все они обеспечивают различную степень повышения производительности обнаружения по сравнению с первичным R-CNN и делают более достижимым точное обнаружение объектов в реальном времени.

На основе базовой архитектуры CNN обнаружение общих объектов достигается с помощью регрессии ограничивающего прямоугольника, в то время как обнаружение заметных объектов выполняется с помощью локального повышения контрастности и сегментации на уровне пикселей. Обнаружение лиц и пешеходов тесно связаны с обычным обнаружением объектов и в основном выполняются с помощью много-

масштабной адаптации и многофункционального слияния/локального повышения контрастности соответственно. Пунктирные линии показывают, что соответствующие домены связаны друг с другом при определенных условиях. Стоит отметить, что охватываемые домены очень разнообразны. Изображения пешеходов и лиц имеют правильную структуру, в то время как изображения общих объектов и сцен имеют более сложные вариации геометрической структуры и компоновки. Поэтому для разных изображений требуются разные глубинные модели.

В настоящее время существует множество библиотек и фреймворков, реализующих алгоритмы глубокого обучения, которые позволяют решить такие подзадачи, такие как окклюзия, беспорядок и низкое разрешение.

Cafe. Популярная библиотека для свёрточных нейронных сетей. Разработана Центром машинного зрения и обучения университета Беркли Заявлена высокая скорость работы. Исполняется как на CPU, так и на GPU. Алгоритм написан на C++, но имеет оболочки на Python и Matlab. Лицензия Беркли [2].

Deeplearning4j. Библиотека на Java и Scala использующая фреймворк с открытым исходным кодом для реализации распределённой обработки неструктурированных и слабоструктурированных данных Apache Spark. Является библиотекой для глубинного обучения общего назначения, предназначенная для запуска на JVM окружении. Ядром библиотеки являются блок научных вычислений написанный на C++. Позволяет создавать слои с заданными параметрами. Интегрирован в пакеты Hadoop и Kafka [2].

DeepLearning-HS. Библиотека глубинного обучения на языке Haskell, поддерживающая распределённые вычисления на технологии CUDA [2].

MatConvNet. Реализация свёрточных нейронных сетей в MATLAB. Заявляется разработчиками как самый быстрый фреймворк для свёрточных нейронных сетей и глубинного обучения с поддержкой вычислений на GPU и CPU. Фронтенд выполнен на языке Python, в то время как сами алгоритмы реализованы на специально разработанном шейдерном ассемблере. Разработан компанией Nervana Systems, которая была куплена Intel [2].

TensorFlow. Библиотека от Google, распространяемая по лицензии Apache 2.0. Поддерживает вычисления на CPU, GPU и специально разработанными компанией Google TPU(Tensor processing units). Фронтенд на Python. Одна из самых популярных, хорошо документированных и развитых библиотек [2].

Theano. Библиотека глубинного обучения для Python с API (в большей части) совместимым с популярной библиотекой NumPy. Позволяет пользователю писать символические формальные математические выражения, из которых автоматически генерируется производный код. Таким образом пользователю не требуется программировать градиенты или обратное распространение ошибки. Такие выражение автоматический компилируются в шейдерный код для CUDA для оптимизации вычислений на GPU [2].

Наиболее популярной является библиотека TensorFlow, т.к. она обладает простотой запуска оболочки на мобильных устройствах, оптимизирована под GPU и достаточно популярна.

**Заключение.** В этой статье представлен подробный обзор основ обнаружения объектов на основе глубокого обучения, которые решают различные подзадачи, такие как окклюзия, беспорядок и низкое разрешение, с различной степенью модификаций в R-CNN.

#### Список цитируемых источников

1. *LeCun, Y.* Deep learning / Y. LeCun, Y. Bengio, G. Hinton. // Nature, vol. 521, no. 7553 — pp. 436—444, 2015.
2. *Шапович, Е. Г.* Автоматизированная классификация диатомовых водорослей / Е. Г. Шапович // Информационные технологии в моделировании и управлении : подходы, методы, решения : сб. науч. ст. II Всерос. науч. конф. с междунар. участием : 22—24 апр. 2019 г. : в 2 ч. — Тольятти, 2019. — Ч. 2 : Материалы секций III, IV. — С. 313—317.